

SAND2011-xxxx  
Unlimited Release  
Updated September 13, 2011

# DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis

## Version 5.1+ Theory Manual

**Brian M. Adams, Keith R. Dalbey, Michael S. Eldred, Laura P. Swiler**  
Optimization and Uncertainty Quantification Department

**William J. Bohnhoff**  
Radiation Transport Department

**John P. Eddy**  
System Readiness and Sustainment Technologies Department

Sandia National Laboratories  
P.O. Box 5800  
Albuquerque, New Mexico 87185

**Patricia D. Hough, Sophia Lefantzi**  
Quantitative Modeling and Analysis Department

Sandia National Laboratories  
P.O. Box 969  
Livermore, CA 94551

## Abstract

The DAKOTA (Design Analysis Kit for Optimization and Terascale Applications) toolkit provides a flexible and extensible interface between simulation codes and iterative analysis methods. DAKOTA contains algorithms for optimization with gradient and nongradient-based methods; uncertainty quantification with sampling, reliability, and stochastic expansion methods; parameter estimation with nonlinear least squares methods; and sensitivity/variance analysis with design of experiments and parameter study methods. These capabilities may be used on their own or as components within advanced strategies such as surrogate-based optimization, mixed integer nonlinear programming, or optimization under uncertainty. By employing object-oriented design to implement abstractions of the key components required for iterative systems analyses, the DAKOTA toolkit provides a flexible and extensible problem-solving environment for design and performance analysis of computational models on high performance computers.

This report serves as a theoretical manual for selected algorithms implemented within the DAKOTA software. It is not intended as a comprehensive theoretical treatment, since a number of existing texts cover general optimization theory, statistical analysis, and other introductory topics. Rather, this manual is intended to summarize a set of DAKOTA-related research publications in the areas of surrogate-based optimization, uncertainty quantification, and optimization under uncertainty that provide the foundation for many of DAKOTA's iterative analysis capabilities.

# Contents

<b>1</b>	<b>Reliability Methods</b>	<b>7</b>
1.1	Local Reliability Methods	7
1.1.1	Mean Value	7
1.1.2	MPP Search Methods	8
1.1.2.1	Limit state approximations	9
1.1.2.2	Probability integrations	11
1.1.2.3	Hessian approximations	12
1.1.2.4	Optimization algorithms	13
1.1.2.5	Warm Starting of MPP Searches	13
1.2	Global Reliability Methods	13
1.2.1	Importance Sampling	13
1.2.2	Efficient Global Optimization	14
1.2.2.1	Gaussian Process Model	15
1.2.2.2	Expected Improvement Function	16
1.2.2.3	Expected Feasibility Function	17
<b>2</b>	<b>Stochastic Expansion Methods</b>	<b>19</b>
2.1	Orthogonal polynomials in the Askey scheme	19
2.2	Numerically generated orthogonal polynomials	20
2.3	Interpolation polynomials	20
2.4	Generalized Polynomial Chaos	20
2.4.1	Expansion truncation and tailoring	21
2.5	Stochastic Collocation	22
2.6	Transformations to uncorrelated standard variables	23
2.7	Spectral projection	24

2.7.1	Sampling	25
2.7.2	Tensor product quadrature	25
2.7.3	Smolyak sparse grids	26
2.7.4	Cubature	27
2.8	Linear regression	28
2.9	Analytic moments	28
2.10	Local sensitivity analysis: derivatives with respect to expansion variables	29
2.11	Global sensitivity analysis: variance-based decomposition	30
2.12	Automated Refinement	30
2.12.1	Uniform p-refinement with isotropic grids	30
2.12.2	Adaptive p-refinement with anisotropic grids	31
2.12.3	Goal-oriented p-refinement with generalized sparse grids	31
<b>3</b>	<b>Epistemic Methods</b>	<b>33</b>
3.1	Dempster-Shafer theory of evidence (DSTE)	33
<b>4</b>	<b>Surrogate-Based Local Minimization</b>	<b>35</b>
4.1	Iterate acceptance logic	37
4.2	Merit functions	38
4.3	Convergence assessment	39
4.4	Constraint relaxation	39
<b>5</b>	<b>Optimization Under Uncertainty (OUU)</b>	<b>43</b>
5.1	Reliability-Based Design Optimization (RBDO)	43
5.1.1	Bi-level RBDO	43
5.1.2	Sequential/Surrogate-based RBDO	45
5.2	Stochastic Expansion-Based Design Optimization (SEBDO)	45
5.2.1	Stochastic Sensitivity Analysis	45
5.2.1.1	Local sensitivity analysis: first-order probabilistic expansions	46
5.2.1.2	Local sensitivity analysis: zeroth-order combined expansions	47
5.2.1.3	Inputs and outputs	47
5.2.2	Optimization Formulations	48
5.2.2.1	Bi-level SEBDO	48
5.2.2.2	Sequential/Surrogate-Based SEBDO	49
5.2.2.3	Multifidelity SEBDO	49

# Chapter 1

## Reliability Methods

### 1.1 Local Reliability Methods

Local reliability methods include the Mean Value method and the family of most probable point (MPP) search methods. Each of these methods is gradient-based, employing local approximations and/or local optimization methods.

#### 1.1.1 Mean Value

The Mean Value method (MV, also known as MVFOSM in [38]) is the simplest, least-expensive reliability method because it estimates the response means, response standard deviations, and all CDF/CCDF response-probability-reliability levels from a single evaluation of response functions and their gradients at the uncertain variable means. This approximation can have acceptable accuracy when the response functions are nearly linear and their distributions are approximately Gaussian, but can have poor accuracy in other situations. The expressions for approximate response mean  $\mu_g$ , approximate response variance  $\sigma_g^2$ , response target to approximate probability/reliability level mapping ( $\bar{z} \rightarrow p, \beta$ ), and probability/reliability target to approximate response level mapping ( $\bar{p}, \bar{\beta} \rightarrow z$ ) are

$$\mu_g = g(\mu_{\mathbf{x}}) \quad (1.1)$$

$$\sigma_g^2 = \sum_i \sum_j Cov(i, j) \frac{dg}{dx_i}(\mu_{\mathbf{x}}) \frac{dg}{dx_j}(\mu_{\mathbf{x}}) \quad (1.2)$$

$$\bar{z} \rightarrow \beta : \quad \beta_{\text{CDF}} = \frac{\mu_g - \bar{z}}{\sigma_g}, \quad \beta_{\text{CCDF}} = \frac{\bar{z} - \mu_g}{\sigma_g} \quad (1.3)$$

$$\bar{\beta} \rightarrow z : \quad z = \mu_g - \sigma_g \bar{\beta}_{\text{CDF}}, \quad z = \mu_g + \sigma_g \bar{\beta}_{\text{CCDF}} \quad (1.4)$$

respectively, where  $\mathbf{x}$  are the uncertain values in the space of the original uncertain variables (“x-space”),  $g(\mathbf{x})$  is the limit state function (the response function for which probability-response level pairs are needed), and  $\beta_{\text{CDF}}$  and  $\beta_{\text{CCDF}}$  are the CDF and CCDF reliability indices, respectively.

With the introduction of second-order limit state information, MVSOSM calculates a second-order mean as

$$\mu_g = g(\mu_{\mathbf{x}}) + \frac{1}{2} \sum_i \sum_j Cov(i, j) \frac{d^2 g}{dx_i dx_j}(\mu_{\mathbf{x}}) \quad (1.5)$$

This is commonly combined with a first-order variance (Equation 1.2), since second-order variance involves higher order distribution moments (skewness, kurtosis) [38] which are often unavailable.

The first-order CDF probability  $p(g \leq z)$ , first-order CCDF probability  $p(g > z)$ ,  $\beta_{\text{CDF}}$ , and  $\beta_{\text{CCDF}}$  are related to one another through

$$p(g \leq z) = \Phi(-\beta_{\text{CDF}}) \quad (1.6)$$

$$p(g > z) = \Phi(-\beta_{\text{CCDF}}) \quad (1.7)$$

$$\beta_{\text{CDF}} = -\Phi^{-1}(p(g \leq z)) \quad (1.8)$$

$$\beta_{\text{CCDF}} = -\Phi^{-1}(p(g > z)) \quad (1.9)$$

$$\beta_{\text{CDF}} = -\beta_{\text{CCDF}} \quad (1.10)$$

$$p(g \leq z) = 1 - p(g > z) \quad (1.11)$$

where  $\Phi()$  is the standard normal cumulative distribution function. A common convention in the literature is to define  $g$  in such a way that the CDF probability for a response level  $z$  of zero (i.e.,  $p(g \leq 0)$ ) is the response metric of interest. DAKOTA is not restricted to this convention and is designed to support CDF or CCDF mappings for general response, probability, and reliability level sequences.

With the Mean Value method, it is possible to obtain importance factors indicating the relative importance of input variables. The importance factors can be viewed as an extension of linear sensitivity analysis combining deterministic gradient information with input uncertainty information, *i.e.* input variable standard deviations. The accuracy of the importance factors is contingent of the validity of the linear approximation used to approximate the true response functions. The importance factors are determined as:

$$ImpFactor_i = \left( \frac{\sigma_{x_i}}{\sigma_g} \frac{dg}{dx_i}(\mu_{\mathbf{x}}) \right)^2 \quad (1.12)$$

### 1.1.2 MPP Search Methods

All other local reliability methods solve an equality-constrained nonlinear optimization problem to compute a most probable point (MPP) and then integrate about this point to compute probabilities. The MPP search is performed in uncorrelated standard normal space (“u-space”) since it simplifies the probability integration: the distance of the MPP from the origin has the meaning of the number of input standard deviations separating the mean response from a particular response threshold. The transformation from correlated non-normal distributions (x-space) to uncorrelated standard normal distributions (u-space) is denoted as  $\mathbf{u} = T(\mathbf{x})$  with the reverse transformation denoted as  $\mathbf{x} = T^{-1}(\mathbf{u})$ . These transformations are nonlinear in general, and possible approaches include the Rosenblatt [59], Nataf [17], and Box-Cox [8] transformations. The nonlinear transformations may also be linearized, and common approaches for this include the Rackwitz-Fiessler [55] two-parameter equivalent normal and the Chen-Lind [12] and Wu-Wirsching [73] three-parameter equivalent normals. DAKOTA employs the Nataf nonlinear transformation which is suitable for the common case when marginal distributions and a correlation matrix are provided, but full joint distributions are not known<sup>1</sup>. This transformation occurs in the following two steps. To transform between the original correlated x-space variables and correlated standard normals (“z-space”), a CDF matching condition is applied for each of the marginal distributions:

$$\Phi(z_i) = F(x_i) \quad (1.13)$$

where  $F()$  is the cumulative distribution function of the original probability distribution. Then, to transform between correlated z-space variables and uncorrelated u-space variables, the Cholesky factor  $\mathbf{L}$  of a modified

<sup>1</sup> If joint distributions are known, then the Rosenblatt transformation is preferred.

correlation matrix is used:

$$\mathbf{z} = \mathbf{L}\mathbf{u} \quad (1.14)$$

where the original correlation matrix for non-normals in x-space has been modified to represent the corresponding “warped” correlation in z-space [17].

The forward reliability analysis algorithm of computing CDF/CCDF probability/reliability levels for specified response levels is called the reliability index approach (RIA), and the inverse reliability analysis algorithm of computing response levels for specified CDF/CCDF probability/reliability levels is called the performance measure approach (PMA) [66]. The differences between the RIA and PMA formulations appear in the objective function and equality constraint formulations used in the MPP searches. For RIA, the MPP search for achieving the specified response level  $\bar{z}$  is formulated as computing the minimum distance in u-space from the origin to the  $\bar{z}$  contour of the limit state response function:

$$\begin{aligned} &\text{minimize} && \mathbf{u}^T \mathbf{u} \\ &\text{subject to} && G(\mathbf{u}) = \bar{z} \end{aligned} \quad (1.15)$$

and for PMA, the MPP search for achieving the specified reliability/probability level  $\bar{\beta}, \bar{p}$  is formulated as computing the minimum/maximum response function value corresponding to a prescribed distance from the origin in u-space:

$$\begin{aligned} &\text{minimize} && \pm G(\mathbf{u}) \\ &\text{subject to} && \mathbf{u}^T \mathbf{u} = \bar{\beta}^2 \end{aligned} \quad (1.16)$$

where  $\mathbf{u}$  is a vector centered at the origin in u-space and  $g(\mathbf{x}) \equiv G(\mathbf{u})$  by definition. In the RIA case, the optimal MPP solution  $\mathbf{u}^*$  defines the reliability index from  $\beta = \pm \|\mathbf{u}^*\|_2$ , which in turn defines the CDF/CCDF probabilities (using Equations 1.6-1.7 in the case of first-order integration). The sign of  $\beta$  is defined by

$$G(\mathbf{u}^*) > G(\mathbf{0}) : \beta_{\text{CDF}} < 0, \beta_{\text{CCDF}} > 0 \quad (1.17)$$

$$G(\mathbf{u}^*) < G(\mathbf{0}) : \beta_{\text{CDF}} > 0, \beta_{\text{CCDF}} < 0 \quad (1.18)$$

where  $G(\mathbf{0})$  is the median limit state response computed at the origin in u-space<sup>2</sup> (where  $\beta_{\text{CDF}} = \beta_{\text{CCDF}} = 0$  and first-order  $p(g \leq z) = p(g > z) = 0.5$ ). In the PMA case, the sign applied to  $G(\mathbf{u})$  (equivalent to minimizing or maximizing  $G(\mathbf{u})$ ) is similarly defined by  $\bar{\beta}$

$$\bar{\beta}_{\text{CDF}} < 0, \bar{\beta}_{\text{CCDF}} > 0 : \text{maximize } G(\mathbf{u}) \quad (1.19)$$

$$\bar{\beta}_{\text{CDF}} > 0, \bar{\beta}_{\text{CCDF}} < 0 : \text{minimize } G(\mathbf{u}) \quad (1.20)$$

and the limit state at the MPP ( $G(\mathbf{u}^*)$ ) defines the desired response level result.

### 1.1.2.1 Limit state approximations

There are a variety of algorithmic variations that are available for use within RIA/PMA reliability analyses. First, one may select among several different limit state approximations that can be used to reduce computational expense during the MPP searches. Local, multipoint, and global approximations of the limit state are possible. [20] investigated local first-order limit state approximations, and [21] investigated local second-order and multipoint approximations. These techniques include:

<sup>2</sup>It is not necessary to explicitly compute the median response since the sign of the inner product  $\langle \mathbf{u}^*, \nabla_{\mathbf{u}} G \rangle$  can be used to determine the orientation of the optimal response with respect to the median response.

1. a single Taylor series per response/reliability/probability level in  $\mathbf{x}$ -space centered at the uncertain variable means. The first-order approach is commonly known as the Advanced Mean Value (AMV) method:

$$g(\mathbf{x}) \cong g(\mu_{\mathbf{x}}) + \nabla_{\mathbf{x}} g(\mu_{\mathbf{x}})^T (\mathbf{x} - \mu_{\mathbf{x}}) \quad (1.21)$$

and the second-order approach has been named AMV<sup>2</sup>:

$$g(\mathbf{x}) \cong g(\mu_{\mathbf{x}}) + \nabla_{\mathbf{x}} g(\mu_{\mathbf{x}})^T (\mathbf{x} - \mu_{\mathbf{x}}) + \frac{1}{2} (\mathbf{x} - \mu_{\mathbf{x}})^T \nabla_{\mathbf{x}}^2 g(\mu_{\mathbf{x}}) (\mathbf{x} - \mu_{\mathbf{x}}) \quad (1.22)$$

2. same as AMV/AMV<sup>2</sup>, except that the Taylor series is expanded in  $\mathbf{u}$ -space. The first-order option has been termed the  $\mathbf{u}$ -space AMV method:

$$G(\mathbf{u}) \cong G(\mu_{\mathbf{u}}) + \nabla_{\mathbf{u}} G(\mu_{\mathbf{u}})^T (\mathbf{u} - \mu_{\mathbf{u}}) \quad (1.23)$$

where  $\mu_{\mathbf{u}} = T(\mu_{\mathbf{x}})$  and is nonzero in general, and the second-order option has been named the  $\mathbf{u}$ -space AMV<sup>2</sup> method:

$$G(\mathbf{u}) \cong G(\mu_{\mathbf{u}}) + \nabla_{\mathbf{u}} G(\mu_{\mathbf{u}})^T (\mathbf{u} - \mu_{\mathbf{u}}) + \frac{1}{2} (\mathbf{u} - \mu_{\mathbf{u}})^T \nabla_{\mathbf{u}}^2 G(\mu_{\mathbf{u}}) (\mathbf{u} - \mu_{\mathbf{u}}) \quad (1.24)$$

3. an initial Taylor series approximation in  $\mathbf{x}$ -space at the uncertain variable means, with iterative expansion updates at each MPP estimate ( $\mathbf{x}^*$ ) until the MPP converges. The first-order option is commonly known as AMV+:

$$g(\mathbf{x}) \cong g(\mathbf{x}^*) + \nabla_{\mathbf{x}} g(\mathbf{x}^*)^T (\mathbf{x} - \mathbf{x}^*) \quad (1.25)$$

and the second-order option has been named AMV<sup>2</sup>+::

$$g(\mathbf{x}) \cong g(\mathbf{x}^*) + \nabla_{\mathbf{x}} g(\mathbf{x}^*)^T (\mathbf{x} - \mathbf{x}^*) + \frac{1}{2} (\mathbf{x} - \mathbf{x}^*)^T \nabla_{\mathbf{x}}^2 g(\mathbf{x}^*) (\mathbf{x} - \mathbf{x}^*) \quad (1.26)$$

4. same as AMV+/AMV<sup>2</sup>+, except that the expansions are performed in  $\mathbf{u}$ -space. The first-order option has been termed the  $\mathbf{u}$ -space AMV+ method.

$$G(\mathbf{u}) \cong G(\mathbf{u}^*) + \nabla_{\mathbf{u}} G(\mathbf{u}^*)^T (\mathbf{u} - \mathbf{u}^*) \quad (1.27)$$

and the second-order option has been named the  $\mathbf{u}$ -space AMV<sup>2</sup>+ method:

$$G(\mathbf{u}) \cong G(\mathbf{u}^*) + \nabla_{\mathbf{u}} G(\mathbf{u}^*)^T (\mathbf{u} - \mathbf{u}^*) + \frac{1}{2} (\mathbf{u} - \mathbf{u}^*)^T \nabla_{\mathbf{u}}^2 G(\mathbf{u}^*) (\mathbf{u} - \mathbf{u}^*) \quad (1.28)$$

5. a multipoint approximation in  $\mathbf{x}$ -space. This approach involves a Taylor series approximation in intermediate variables where the powers used for the intermediate variables are selected to match information at the current and previous expansion points. Based on the two-point exponential approximation concept (TPEA, [27]), the two-point adaptive nonlinearity approximation (TANA-3, [78]) approximates the limit state as:

$$g(\mathbf{x}) \cong g(\mathbf{x}_2) + \sum_{i=1}^n \frac{\partial g}{\partial x_i}(\mathbf{x}_2) \frac{x_{i,2}^{1-p_i}}{p_i} (x_i^{p_i} - x_{i,2}^{p_i}) + \frac{1}{2} \epsilon(\mathbf{x}) \sum_{i=1}^n (x_i^{p_i} - x_{i,2}^{p_i})^2 \quad (1.29)$$

where  $n$  is the number of uncertain variables and:

$$p_i = 1 + \ln \left[ \frac{\frac{\partial g}{\partial x_i}(\mathbf{x}_1)}{\frac{\partial g}{\partial x_i}(\mathbf{x}_2)} \right] \bigg/ \ln \left[ \frac{x_{i,1}}{x_{i,2}} \right] \quad (1.30)$$

$$\epsilon(\mathbf{x}) = \frac{H}{\sum_{i=1}^n (x_i^{p_i} - x_{i,1}^{p_i})^2 + \sum_{i=1}^n (x_i^{p_i} - x_{i,2}^{p_i})^2} \quad (1.31)$$

$$H = 2 \left[ g(\mathbf{x}_1) - g(\mathbf{x}_2) - \sum_{i=1}^n \frac{\partial g}{\partial x_i}(\mathbf{x}_2) \frac{x_{i,2}^{1-p_i}}{p_i} (x_{i,1}^{p_i} - x_{i,2}^{p_i}) \right] \quad (1.32)$$



and  $\mathbf{x}_2$  and  $\mathbf{x}_1$  are the current and previous MPP estimates in x-space, respectively. Prior to the availability of two MPP estimates, x-space AMV+ is used.

6. a multipoint approximation in u-space. The u-space TANA-3 approximates the limit state as:

$$G(\mathbf{u}) \cong G(\mathbf{u}_2) + \sum_{i=1}^n \frac{\partial G}{\partial u_i}(\mathbf{u}_2) \frac{u_{i,2}^{1-p_i}}{p_i} (u_i^{p_i} - u_{i,2}^{p_i}) + \frac{1}{2} \epsilon(\mathbf{u}) \sum_{i=1}^n (u_i^{p_i} - u_{i,2}^{p_i})^2 \quad (1.33)$$

where:

$$p_i = 1 + \ln \left[ \frac{\frac{\partial G}{\partial u_i}(\mathbf{u}_1)}{\frac{\partial G}{\partial u_i}(\mathbf{u}_2)} \right] / \ln \left[ \frac{u_{i,1}}{u_{i,2}} \right] \quad (1.34)$$

$$\epsilon(\mathbf{u}) = \frac{H}{\sum_{i=1}^n (u_i^{p_i} - u_{i,1}^{p_i})^2 + \sum_{i=1}^n (u_i^{p_i} - u_{i,2}^{p_i})^2} \quad (1.35)$$

$$H = 2 \left[ G(\mathbf{u}_1) - G(\mathbf{u}_2) - \sum_{i=1}^n \frac{\partial G}{\partial u_i}(\mathbf{u}_2) \frac{u_{i,2}^{1-p_i}}{p_i} (u_{i,1}^{p_i} - u_{i,2}^{p_i}) \right] \quad (1.36)$$

and  $\mathbf{u}_2$  and  $\mathbf{u}_1$  are the current and previous MPP estimates in u-space, respectively. Prior to the availability of two MPP estimates, u-space AMV+ is used.

7. the MPP search on the original response functions without the use of any approximations. Combining this option with first-order and second-order integration approaches (see next section) results in the traditional first-order and second-order reliability methods (FORM and SORM).

The Hessian matrices in AMV<sup>2</sup> and AMV<sup>2</sup>+ may be available analytically, estimated numerically, or approximated through quasi-Newton updates. The selection between x-space or u-space for performing approximations depends on where the approximation will be more accurate, since this will result in more accurate MPP estimates (AMV, AMV<sup>2</sup>) or faster convergence (AMV+, AMV<sup>2</sup>+, TANA). Since this relative accuracy depends on the forms of the limit state  $g(x)$  and the transformation  $T(x)$  and is therefore application dependent in general, DAKOTA supports both options. A concern with approximation-based iterative search methods (i.e., AMV+, AMV<sup>2</sup>+ and TANA) is the robustness of their convergence to the MPP. It is possible for the MPP iterates to oscillate or even diverge. However, to date, this occurrence has been relatively rare, and DAKOTA contains checks that monitor for this behavior. Another concern with TANA is numerical safeguarding (e.g., the possibility of raising negative  $x_i$  or  $u_i$  values to nonintegral  $p_i$  exponents in Equations 1.29, 1.31-1.33, and 1.35-1.36). Safeguarding involves offsetting negative  $x_i$  or  $u_i$  and, for potential numerical difficulties with the logarithm ratios in Equations 1.30 and 1.34, reverting to either the linear ( $p_i = 1$ ) or reciprocal ( $p_i = -1$ ) approximation based on which approximation has lower error in  $\frac{\partial g}{\partial x_i}(\mathbf{x}_1)$  or  $\frac{\partial G}{\partial u_i}(\mathbf{u}_1)$ .

### 1.1.2.2 Probability integrations

The second algorithmic variation involves the integration approach for computing probabilities at the MPP, which can be selected to be first-order (Equations 1.6-1.7) or second-order integration. Second-order integration involves applying a curvature correction [9, 40, 41]. Breitung applies a correction based on asymptotic analysis [9]:

$$p = \Phi(-\beta_p) \prod_{i=1}^{n-1} \frac{1}{\sqrt{1 + \beta_p \kappa_i}} \quad (1.37)$$

where  $\kappa_i$  are the principal curvatures of the limit state function (the eigenvalues of an orthonormal transformation of  $\nabla_{\mathbf{u}}^2 G$ , taken positive for a convex limit state) and  $\beta_p \geq 0$  (a CDF or CCDF probability correction is selected to

obtain the correct sign for  $\beta_p$ ). An alternate correction in [40] is consistent in the asymptotic regime ( $\beta_p \rightarrow \infty$ ) but does not collapse to first-order integration for  $\beta_p = 0$ :

$$p = \Phi(-\beta_p) \prod_{i=1}^{n-1} \frac{1}{\sqrt{1 + \psi(-\beta_p)\kappa_i}} \quad (1.38)$$

where  $\psi() = \frac{\phi()}{\Phi()}$  and  $\phi()$  is the standard normal density function. [41] applies further corrections to Equation 1.38 based on point concentration methods. At this time, all three approaches are available within the code, but the Hohenbichler-Rackwitz correction is used by default (switching the correction is a compile-time option in the source code and has not currently been exposed in the input specification).

### 1.1.2.3 Hessian approximations

To use a second-order Taylor series or a second-order integration when second-order information ( $\nabla_{\mathbf{x}}^2 g$ ,  $\nabla_{\mathbf{u}}^2 G$ , and/or  $\kappa$ ) is not directly available, one can estimate the missing information using finite differences or approximate it through use of quasi-Newton approximations. These procedures will often be needed to make second-order approaches practical for engineering applications.

In the finite difference case, numerical Hessians are commonly computed using either first-order forward differences of gradients using

$$\nabla^2 g(\mathbf{x}) \cong \frac{\nabla g(\mathbf{x} + h\mathbf{e}_i) - \nabla g(\mathbf{x})}{h} \quad (1.39)$$

to estimate the  $i^{th}$  Hessian column when gradients are analytically available, or second-order differences of function values using

$$\nabla^2 g(\mathbf{x}) \cong \frac{g(\mathbf{x}+h\mathbf{e}_i+h\mathbf{e}_j)-g(\mathbf{x}+h\mathbf{e}_i-h\mathbf{e}_j)-g(\mathbf{x}-h\mathbf{e}_i+h\mathbf{e}_j)+g(\mathbf{x}-h\mathbf{e}_i-h\mathbf{e}_j)}{4h^2} \quad (1.40)$$

to estimate the  $ij^{th}$  Hessian term when gradients are not directly available. This approach has the advantage of locally-accurate Hessians for each point of interest (which can lead to quadratic convergence rates in discrete Newton methods), but has the disadvantage that numerically estimating each of the matrix terms can be expensive.

Quasi-Newton approximations, on the other hand, do not reevaluate all of the second-order information for every point of interest. Rather, they accumulate approximate curvature information over time using secant updates. Since they utilize the existing gradient evaluations, they do not require any additional function evaluations for evaluating the Hessian terms. The quasi-Newton approximations of interest include the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update

$$\mathbf{B}_{k+1} = \mathbf{B}_k - \frac{\mathbf{B}_k \mathbf{s}_k \mathbf{s}_k^T \mathbf{B}_k}{\mathbf{s}_k^T \mathbf{B}_k \mathbf{s}_k} + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k} \quad (1.41)$$

which yields a sequence of symmetric positive definite Hessian approximations, and the Symmetric Rank 1 (SR1) update

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{(\mathbf{y}_k - \mathbf{B}_k \mathbf{s}_k)(\mathbf{y}_k - \mathbf{B}_k \mathbf{s}_k)^T}{(\mathbf{y}_k - \mathbf{B}_k \mathbf{s}_k)^T \mathbf{s}_k} \quad (1.42)$$

which yields a sequence of symmetric, potentially indefinite, Hessian approximations.  $\mathbf{B}_k$  is the  $k^{th}$  approximation to the Hessian  $\nabla^2 g$ ,  $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$  is the step and  $\mathbf{y}_k = \nabla g_{k+1} - \nabla g_k$  is the corresponding yield in the gradients. The selection of BFGS versus SR1 involves the importance of retaining positive definiteness in the Hessian approximations; if the procedure does not require it, then the SR1 update can be more accurate if the true Hessian is not positive definite. Initial scalings for  $\mathbf{B}_0$  and numerical safeguarding techniques (damped BFGS, update skipping) are described in [21].

### 1.1.2.4 Optimization algorithms

The next algorithmic variation involves the optimization algorithm selection for solving Eqs. 1.15 and 1.16. The Hasofer-Lind Rackwitz-Fissler (HL-RF) algorithm [38] is a classical approach that has been broadly applied. It is a Newton-based approach lacking line search/trust region globalization, and is generally regarded as computationally efficient but occasionally unreliable. DAKOTA takes the approach of employing robust, general-purpose optimization algorithms with provable convergence properties. In particular, we employ the sequential quadratic programming (SQP) and nonlinear interior-point (NIP) optimization algorithms from the NPSOL [34] and OPT++ [47] libraries, respectively.

### 1.1.2.5 Warm Starting of MPP Searches

The final algorithmic variation for local reliability methods involves the use of warm starting approaches for improving computational efficiency. [20] describes the acceleration of MPP searches through warm starting with approximate iteration increment, with  $z/p/\beta$  level increment, and with design variable increment. Warm started data includes the expansion point and associated response values and the MPP optimizer initial guess. Projections are used when an increment in  $z/p/\beta$  level or design variables occurs. Warm starts were consistently effective in [20], with greater effectiveness for smaller parameter changes, and are used by default in DAKOTA.

## 1.2 Global Reliability Methods

Local reliability methods, while computationally efficient, have well-known failure mechanisms. When confronted with a limit state function that is nonsmooth, local gradient-based optimizers may stall due to gradient inaccuracy and fail to converge to an MPP. Moreover, if the limit state is multimodal (multiple MPPs), then a gradient-based local method can, at best, locate only one local MPP solution. Finally, a linear (Eqs. 1.6–1.7) or parabolic (Eqs. 1.37–1.38) approximation to the limit state at this MPP may fail to adequately capture the contour of a highly nonlinear limit state.

A reliability analysis method that is both efficient when applied to expensive response functions and accurate for a response function of any arbitrary shape is needed. This section develops such a method based on efficient global optimization [44] (EGO) to the search for multiple points on or near the limit state throughout the random variable space. By locating multiple points on the limit state, more complex limit states can be accurately modeled, resulting in a more accurate assessment of the reliability. It should be emphasized here that these multiple points exist on a single limit state. Because of its roots in efficient global optimization, this method of reliability analysis is called efficient global reliability analysis (EGRA) [7]. The following two subsections describe two capabilities that are incorporated into the EGRA algorithm: importance sampling and EGO.

### 1.2.1 Importance Sampling

An alternative to MPP search methods is to directly perform the probability integration numerically by sampling the response function. Sampling methods do not rely on a simplifying approximation to the shape of the limit state, so they can be more accurate than FORM and SORM, but they can also be prohibitively expensive because they generally require a large number of response function evaluations. Importance sampling methods reduce this expense by focusing the samples in the important regions of the uncertain space. They do this by centering the sampling density function at the MPP rather than at the mean. This ensures the samples will lie the region of interest, thus increasing the efficiency of the sampling method. Adaptive importance sampling (AIS) further improves the efficiency by adaptively updating the sampling density function. Multimodal adaptive importance

sampling [18, 80] is a variation of AIS that allows for the use of multiple sampling densities making it better suited for cases where multiple sections of the limit state are highly probable.

Note that importance sampling methods require that the location of at least one MPP be known because it is used to center the initial sampling density. However, current gradient-based, local search methods used in MPP search may fail to converge or may converge to poor solutions for highly nonlinear problems, possibly making these methods inapplicable. As the next section describes, EGO is a global optimization method that does not depend on the availability of accurate gradient information, making convergence more reliable for nonsmooth response functions. Moreover, EGO has the ability to locate multiple failure points, which would provide multiple starting points and thus a good multimodal sampling density for the initial steps of multimodal AIS. The resulting Gaussian process model is accurate in the vicinity of the limit state, thereby providing an inexpensive surrogate that can be used to provide response function samples. As will be seen, using EGO to locate multiple points along the limit state, and then using the resulting Gaussian process model to provide function evaluations in multimodal AIS for the probability integration, results in an accurate and efficient reliability analysis tool.

### 1.2.2 Efficient Global Optimization

Efficient Global Optimization (EGO) was developed to facilitate the unconstrained minimization of expensive implicit response functions. The method builds an initial Gaussian process model as a global surrogate for the response function, then intelligently selects additional samples to be added for inclusion in a new Gaussian process model in subsequent iterations. The new samples are selected based on how much they are expected to improve the current best solution to the optimization problem. When this expected improvement is acceptably small, the globally optimal solution has been found. The application of this methodology to equality-constrained reliability analysis is the primary contribution of EGRA.

Efficient global optimization was originally proposed by Jones et al. [44] and has been adapted into similar methods such as sequential kriging optimization (SKO) [43]. The main difference between SKO and EGO lies within the specific formulation of what is known as the expected improvement function (EIF), which is the feature that sets all EGO/SKO-type methods apart from other global optimization methods. The EIF is used to select the location at which a new training point should be added to the Gaussian process model by maximizing the amount of improvement in the objective function that can be expected by adding that point. A point could be expected to produce an improvement in the objective function if its predicted value is better than the current best solution, or if the uncertainty in its prediction is such that the probability of it producing a better solution is high. Because the uncertainty is higher in regions of the design space with fewer observations, this provides a balance between exploiting areas of the design space that predict good solutions, and exploring areas where more information is needed.

The general procedure of these EGO-type methods is:

1. Build an initial Gaussian process model of the objective function.
2. Find the point that maximizes the EIF. If the EIF value at this point is sufficiently small, stop.
3. Evaluate the objective function at the point where the EIF is maximized. Update the Gaussian process model using this new point. Go to Step 2.

The following sections discuss the construction of the Gaussian process model used, the form of the EIF, and then a description of how that EIF is modified for application to reliability analysis.

### 1.2.2.1 Gaussian Process Model

Gaussian process (GP) models are set apart from other surrogate models because they provide not just a predicted value at an unsampled point, but also an estimate of the prediction variance. This variance gives an indication of the uncertainty in the GP model, which results from the construction of the covariance function. This function is based on the idea that when input points are near one another, the correlation between their corresponding outputs will be high. As a result, the uncertainty associated with the model's predictions will be small for input points which are near the points used to train the model, and will increase as one moves further from the training points.

It is assumed that the true response function being modeled  $G(\mathbf{u})$  can be described by: [16]

$$G(\mathbf{u}) = \mathbf{h}(\mathbf{u})^T \boldsymbol{\beta} + Z(\mathbf{u}) \quad (1.43)$$

where  $\mathbf{h}(\cdot)$  is the trend of the model,  $\boldsymbol{\beta}$  is the vector of trend coefficients, and  $Z(\cdot)$  is a stationary Gaussian process with zero mean (and covariance defined below) that describes the departure of the model from its underlying trend. The trend of the model can be assumed to be any function, but taking it to be a constant value has been reported to be generally sufficient. [60] For the work presented here, the trend is assumed constant and  $\boldsymbol{\beta}$  is taken as simply the mean of the responses at the training points. The covariance between outputs of the Gaussian process  $Z(\cdot)$  at points  $\mathbf{a}$  and  $\mathbf{b}$  is defined as:

$$\text{Cov}[Z(\mathbf{a}), Z(\mathbf{b})] = \sigma_Z^2 R(\mathbf{a}, \mathbf{b}) \quad (1.44)$$

where  $\sigma_Z^2$  is the process variance and  $R(\cdot)$  is the correlation function. There are several options for the correlation function, but the squared-exponential function is common [60], and is used here for  $R(\cdot)$ :

$$R(\mathbf{a}, \mathbf{b}) = \exp \left[ - \sum_{i=1}^d \theta_i (a_i - b_i)^2 \right] \quad (1.45)$$

where  $d$  represents the dimensionality of the problem (the number of random variables), and  $\theta_i$  is a scale parameter that indicates the correlation between the points within dimension  $i$ . A large  $\theta_i$  is representative of a short correlation length.

The expected value  $\mu_G(\cdot)$  and variance  $\sigma_G^2(\cdot)$  of the GP model prediction at point  $\mathbf{u}$  are:

$$\mu_G(\mathbf{u}) = \mathbf{h}(\mathbf{u})^T \boldsymbol{\beta} + \mathbf{r}(\mathbf{u})^T \mathbf{R}^{-1} (\mathbf{g} - \mathbf{F} \boldsymbol{\beta}) \quad (1.46)$$

$$\sigma_G^2(\mathbf{u}) = \sigma_Z^2 - [\mathbf{h}(\mathbf{u})^T \quad \mathbf{r}(\mathbf{u})^T] \begin{bmatrix} \mathbf{0} & \mathbf{F}^T \\ \mathbf{F} & \mathbf{R} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{h}(\mathbf{u}) \\ \mathbf{r}(\mathbf{u}) \end{bmatrix} \quad (1.47)$$

where  $\mathbf{r}(\mathbf{u})$  is a vector containing the covariance between  $\mathbf{u}$  and each of the  $n$  training points (defined by Eq. 1.44),  $\mathbf{R}$  is an  $n \times n$  matrix containing the correlation between each pair of training points,  $\mathbf{g}$  is the vector of response outputs at each of the training points, and  $\mathbf{F}$  is an  $n \times q$  matrix with rows  $\mathbf{h}(\mathbf{u}_i)^T$  (the trend function for training point  $i$  containing  $q$  terms; for a constant trend  $q=1$ ). This form of the variance accounts for the uncertainty in the trend coefficients  $\boldsymbol{\beta}$ , but assumes that the parameters governing the covariance function ( $\sigma_Z^2$  and  $\boldsymbol{\theta}$ ) have known values.

The parameters  $\sigma_Z^2$  and  $\boldsymbol{\theta}$  are determined through maximum likelihood estimation. This involves taking the log of the probability of observing the response values  $\mathbf{g}$  given the covariance matrix  $\mathbf{R}$ , which can be written as: [60]

$$\log [p(\mathbf{g}|\mathbf{R})] = -\frac{1}{n} \log |\mathbf{R}| - \log (\hat{\sigma}_Z^2) \quad (1.48)$$

where  $|\mathbf{R}|$  indicates the determinant of  $\mathbf{R}$ , and  $\hat{\sigma}_Z^2$  is the optimal value of the variance given an estimate of  $\boldsymbol{\theta}$  and is defined by:

$$\hat{\sigma}_Z^2 = \frac{1}{n} (\mathbf{g} - \mathbf{F} \boldsymbol{\beta})^T \mathbf{R}^{-1} (\mathbf{g} - \mathbf{F} \boldsymbol{\beta}) \quad (1.49)$$

Maximizing Eq. 1.48 gives the maximum likelihood estimate of  $\boldsymbol{\theta}$ , which in turn defines  $\sigma_Z^2$ .

### 1.2.2.2 Expected Improvement Function

The expected improvement function is used to select the location at which a new training point should be added. The EIF is defined as the expectation that any point in the search space will provide a better solution than the current best solution based on the expected values and variances predicted by the GP model. An important feature of the EIF is that it provides a balance between exploiting areas of the design space where good solutions have been found, and exploring areas of the design space where the uncertainty is high. First, recognize that at any point in the design space, the GP prediction  $\hat{G}(\mathbf{u})$  is a Gaussian distribution:

$$\hat{G}(\mathbf{u}) \sim N[\mu_G(\mathbf{u}), \sigma_G(\mathbf{u})] \quad (1.50)$$

where the mean  $\mu_G(\mathbf{u})$  and the variance  $\sigma_G^2(\mathbf{u})$  were defined in Eqs. 1.46 and 1.47, respectively. The EIF is defined as: [44]

$$EI(\hat{G}(\mathbf{u})) \equiv E \left[ \max \left( G(\mathbf{u}^*) - \hat{G}(\mathbf{u}), 0 \right) \right] \quad (1.51)$$

where  $G(\mathbf{u}^*)$  is the current best solution chosen from among the true function values at the training points (henceforth referred to as simply  $G^*$ ). This expectation can then be computed by integrating over the distribution  $\hat{G}(\mathbf{u})$  with  $G^*$  held constant:

$$EI(\hat{G}(\mathbf{u})) = \int_{-\infty}^{G^*} (G^* - G) \hat{G}(\mathbf{u}) dG \quad (1.52)$$

where  $G$  is a realization of  $\hat{G}$ . This integral can be expressed analytically as: [44]

$$EI(\hat{G}(\mathbf{u})) = (G^* - \mu_G) \Phi \left( \frac{G^* - \mu_G}{\sigma_G} \right) + \sigma_G \phi \left( \frac{G^* - \mu_G}{\sigma_G} \right) \quad (1.53)$$

where it is understood that  $\mu_G$  and  $\sigma_G$  are functions of  $\mathbf{u}$ .

The point at which the EIF is maximized is selected as an additional training point. With the new training point added, a new GP model is built and then used to construct another EIF, which is then used to choose another new training point, and so on, until the value of the EIF at its maximized point is below some specified tolerance. In Ref. [43] this maximization is performed using a Nelder-Mead simplex approach, which is a local optimization method. Because the EIF is often highly multimodal [44] it is expected that Nelder-Mead may fail to converge to the true global optimum. In Ref. [44], a branch-and-bound technique for maximizing the EIF is used, but was found to often be too expensive to run to convergence. In DAKOTA, an implementation of the DIRECT global optimization algorithm is used [30].

It is important to understand how the use of this EIF leads to optimal solutions. Eq. 1.53 indicates how much the objective function value at  $\mathbf{x}$  is expected to be less than the predicted value at the current best solution. Because the GP model provides a Gaussian distribution at each predicted point, expectations can be calculated. Points with good expected values and even a small variance will have a significant expectation of producing a better solution (exploitation), but so will points that have relatively poor expected values and greater variance (exploration).

The application of EGO to reliability analysis, however, is made more complicated due to the inclusion of equality constraints (see Eqs. 1.15-1.16). For inverse reliability analysis, this extra complication is small. The response being modeled by the GP is the objective function of the optimization problem (see Eq. 1.16) and the deterministic constraint might be handled through the use of a merit function, thereby allowing EGO to solve this equality-constrained optimization problem. Here the problem lies in the interpretation of the constraint for multimodal problems as mentioned previously. In the forward reliability case, the response function appears in the constraint rather than the objective. Here, the maximization of the EIF is inappropriate because feasibility is the main concern. This application is therefore a significant departure from the original objective of EGO and requires a new formulation. For this problem, the expected feasibility function is introduced.

### 1.2.2.3 Expected Feasibility Function

The expected improvement function provides an indication of how much the true value of the response at a point can be expected to be less than the current best solution. It therefore makes little sense to apply this to the forward reliability problem where the goal is not to minimize the response, but rather to find where it is equal to a specified threshold value. The expected feasibility function (EFF) is introduced here to provide an indication of how well the true value of the response is expected to satisfy the equality constraint  $G(\mathbf{u}) = \bar{z}$ . Inspired by the contour estimation work in [56], this expectation can be calculated in a similar fashion as Eq. 1.52 by integrating over a region in the immediate vicinity of the threshold value  $\bar{z} \pm \epsilon$ :

$$EF(\hat{G}(\mathbf{u})) = \int_{z-\epsilon}^{z+\epsilon} [\epsilon - |z - G|] \hat{G}(\mathbf{u}) dG \quad (1.54)$$

where  $G$  denotes a realization of the distribution  $\hat{G}$ , as before. Allowing  $z^+$  and  $z^-$  to denote  $\bar{z} \pm \epsilon$ , respectively, this integral can be expressed analytically as:

$$\begin{aligned} EF(\hat{G}(\mathbf{u})) = & (\mu_G - \bar{z}) \left[ 2\Phi\left(\frac{\bar{z} - \mu_G}{\sigma_G}\right) - \Phi\left(\frac{z^- - \mu_G}{\sigma_G}\right) - \Phi\left(\frac{z^+ - \mu_G}{\sigma_G}\right) \right] \\ & - \sigma_G \left[ 2\phi\left(\frac{\bar{z} - \mu_G}{\sigma_G}\right) - \phi\left(\frac{z^- - \mu_G}{\sigma_G}\right) - \phi\left(\frac{z^+ - \mu_G}{\sigma_G}\right) \right] \\ & + \epsilon \left[ \Phi\left(\frac{z^+ - \mu_G}{\sigma_G}\right) - \Phi\left(\frac{z^- - \mu_G}{\sigma_G}\right) \right] \end{aligned} \quad (1.55)$$

where  $\epsilon$  is proportional to the standard deviation of the GP predictor ( $\epsilon \propto \sigma_G$ ). In this case,  $z^-$ ,  $z^+$ ,  $\mu_G$ ,  $\sigma_G$ , and  $\epsilon$  are all functions of the location  $\mathbf{u}$ , while  $\bar{z}$  is a constant. Note that the EFF provides the same balance between exploration and exploitation as is captured in the EIF. Points where the expected value is close to the threshold ( $\mu_G \approx \bar{z}$ ) and points with a large uncertainty in the prediction will have large expected feasibility values.





## Chapter 2

# Stochastic Expansion Methods

This chapter explores two approaches to forming stochastic expansions, the polynomial chaos expansion (PCE) and stochastic collocation (SC). Both approaches capture the functional relationship between a set of output response metrics and a set of input random variables.

### 2.1 Orthogonal polynomials in the Askey scheme

Table 2.1 shows the set of classical orthogonal polynomials which provide an optimal basis for different continuous probability distribution types. It is derived from the family of hypergeometric orthogonal polynomials known as the Askey scheme [5], for which the Hermite polynomials originally employed by Wiener [70] are a subset. The optimality of these basis selections derives from their orthogonality with respect to weighting functions that correspond to the probability density functions (PDFs) of the continuous distributions when placed in a standard form. The density and weighting functions differ by a constant factor due to the requirement that the integral of the PDF over the support range is one.

Table 2.1: Linkage between standard forms of continuous probability distributions and Askey scheme of continuous hyper-geometric polynomials.

Distribution	Density function	Polynomial	Weight function	Support range
Normal	$\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$	Hermite $He_n(x)$	$e^{-\frac{x^2}{2}}$	$[-\infty, \infty]$
Uniform	$\frac{1}{2}$	Legendre $P_n(x)$	1	$[-1, 1]$
Beta	$\frac{(1-x)^\alpha (1+x)^\beta}{2^{\alpha+\beta+1} B(\alpha+1, \beta+1)}$	Jacobi $P_n^{(\alpha, \beta)}(x)$	$(1-x)^\alpha (1+x)^\beta$	$[-1, 1]$
Exponential	$e^{-x}$	Laguerre $L_n(x)$	$e^{-x}$	$[0, \infty]$
Gamma	$\frac{x^\alpha e^{-x}}{\Gamma(\alpha+1)}$	Generalized Laguerre $L_n^{(\alpha)}(x)$	$x^\alpha e^{-x}$	$[0, \infty]$

Note that Legendre is a special case of Jacobi for  $\alpha = \beta = 0$ , Laguerre is a special case of generalized Laguerre for  $\alpha = 0$ ,  $\Gamma(a)$  is the Gamma function which extends the factorial function to continuous values, and  $B(a, b)$  is the Beta function defined as  $B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$ . Some care is necessary when specifying the  $\alpha$  and  $\beta$  parameters for the Jacobi and generalized Laguerre polynomials since the orthogonal polynomial conventions [1] differ from the common statistical PDF conventions. The former conventions are used in Table 2.1.

## 2.2 Numerically generated orthogonal polynomials

If all random inputs can be described using independent normal, uniform, exponential, beta, and gamma distributions, then Askey polynomials can be directly applied. If correlation or other distribution types are present, then additional techniques are required. One solution is to employ nonlinear variable transformations as described in Section 2.6 such that an Askey basis can be applied in the transformed space. This can be effective as shown in [26], but convergence rates are typically degraded. In addition, correlation coefficients are warped by the nonlinear transformation [17], and simple expressions for these transformed correlation values are not always readily available. An alternative is to numerically generate the orthogonal polynomials (using Gauss-Wigert [61], discretized Stieltjes [31], Chebyshev [31], or Gramm-Schmidt [71] approaches) and then compute their Gauss points and weights (using the Golub-Welsch [37] tridiagonal eigensolution). These solutions are optimal for given random variable sets having arbitrary probability density functions and eliminate the need to induce additional nonlinearity through variable transformations, but performing this process for general joint density functions with correlation is a topic of ongoing research (refer to Section 2.6 for additional details).

## 2.3 Interpolation polynomials

Lagrange polynomials interpolate a set of points in a single dimension using the functional form

$$L_j = \prod_{\substack{k=1 \\ k \neq j}}^m \frac{\xi - \xi_k}{\xi_j - \xi_k} \quad (2.1)$$

where it is evident that  $L_j$  is 1 at  $\xi = \xi_j$ , is 0 for each of the points  $\xi = \xi_k$ , and has order  $m - 1$ .

For interpolation of a response function  $R$  in one dimension over  $m$  points, the expression

$$R(\xi) \cong \sum_{j=1}^m r(\xi_j) L_j(\xi) \quad (2.2)$$

reproduces the response values  $r(\xi_j)$  at the interpolation points and smoothly interpolates between these values at other points. For interpolation in multiple dimensions, a tensor-product approach is used wherein

$$R(\boldsymbol{\xi}) \cong \sum_{j_1=1}^{m_{i_1}} \cdots \sum_{j_n=1}^{m_{i_n}} r(\xi_{j_1}^{i_1}, \dots, \xi_{j_n}^{i_n}) (L_{j_1}^{i_1} \otimes \cdots \otimes L_{j_n}^{i_n}) = \sum_{j=1}^{N_p} r_j(\boldsymbol{\xi}) L_j(\boldsymbol{\xi}) \quad (2.3)$$

where  $\mathbf{i} = (m_1, m_2, \dots, m_n)$  are the number of nodes used in the  $n$ -dimensional interpolation and  $\xi_j^i$  is the  $j$ -th point in the  $i$ -th direction. As will be seen later, interpolation on sparse grids involves a summation of these tensor products with varying  $\mathbf{i}$  levels.

## 2.4 Generalized Polynomial Chaos

The set of polynomials from 2.1 and 2.2 are used as an orthogonal basis to approximate the functional form between the stochastic response output and each of its random inputs. The chaos expansion for a response  $R$  takes the form

$$R = a_0 B_0 + \sum_{i_1=1}^{\infty} a_{i_1} B_1(\xi_{i_1}) + \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} a_{i_1 i_2} B_2(\xi_{i_1}, \xi_{i_2}) + \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} \sum_{i_3=1}^{i_2} a_{i_1 i_2 i_3} B_3(\xi_{i_1}, \xi_{i_2}, \xi_{i_3}) + \dots \quad (2.4)$$

where the random vector dimension is unbounded and each additional set of nested summations indicates an additional order of polynomials in the expansion. This expression can be simplified by replacing the order-based indexing with a term-based indexing

$$R = \sum_{j=0}^{\infty} \alpha_j \Psi_j(\boldsymbol{\xi}) \quad (2.5)$$

where there is a one-to-one correspondence between  $a_{i_1 i_2 \dots i_n}$  and  $\alpha_j$  and between  $B_n(\xi_{i_1}, \xi_{i_2}, \dots, \xi_{i_n})$  and  $\Psi_j(\boldsymbol{\xi})$ . Each of the  $\Psi_j(\boldsymbol{\xi})$  are multivariate polynomials which involve products of the one-dimensional polynomials. For example, a multivariate Hermite polynomial  $B(\boldsymbol{\xi})$  of order  $n$  is defined from

$$B_n(\xi_{i_1}, \dots, \xi_{i_n}) = e^{\frac{1}{2}\boldsymbol{\xi}^T \boldsymbol{\xi}} (-1)^n \frac{\partial^n}{\partial \xi_{i_1} \dots \partial \xi_{i_n}} e^{-\frac{1}{2}\boldsymbol{\xi}^T \boldsymbol{\xi}} \quad (2.6)$$

which can be shown to be a product of one-dimensional Hermite polynomials involving a multi-index  $m_i^j$ :

$$B_n(\xi_{i_1}, \dots, \xi_{i_n}) = \Psi_j(\boldsymbol{\xi}) = \prod_{i=1}^n \psi_{m_i^j}(\xi_i) \quad (2.7)$$

In the case of a mixed basis, the same multi-index definition is employed although the one-dimensional polynomials  $\psi_{m_i^j}$  are heterogeneous in type.

### 2.4.1 Expansion truncation and tailoring

In practice, one truncates the infinite expansion at a finite number of random variables and a finite expansion order

$$R \cong \sum_{j=0}^P \alpha_j \Psi_j(\boldsymbol{\xi}) \quad (2.8)$$

Traditionally, the polynomial chaos expansion includes a complete basis of polynomials up to a fixed total-order specification. That is, for an expansion of total order  $p$  involving  $n$  random variables, the multi-index defining the set of  $\Psi_j$  is constrained by

$$\sum_{i=1}^n m_i^j \leq p \quad (2.9)$$

For example, the multidimensional basis polynomials for a second-order expansion over two random dimensions are

$$\begin{aligned} \Psi_0(\boldsymbol{\xi}) &= \psi_0(\xi_1) \psi_0(\xi_2) = 1 \\ \Psi_1(\boldsymbol{\xi}) &= \psi_1(\xi_1) \psi_0(\xi_2) = \xi_1 \\ \Psi_2(\boldsymbol{\xi}) &= \psi_0(\xi_1) \psi_1(\xi_2) = \xi_2 \\ \Psi_3(\boldsymbol{\xi}) &= \psi_2(\xi_1) \psi_0(\xi_2) = \xi_1^2 - 1 \\ \Psi_4(\boldsymbol{\xi}) &= \psi_1(\xi_1) \psi_1(\xi_2) = \xi_1 \xi_2 \\ \Psi_5(\boldsymbol{\xi}) &= \psi_0(\xi_1) \psi_2(\xi_2) = \xi_2^2 - 1 \end{aligned}$$

The total number of terms  $N_t$  in an expansion of total order  $p$  involving  $n$  random variables is given by

$$N_t = 1 + P = 1 + \sum_{s=1}^p \frac{1}{s!} \prod_{r=0}^{s-1} (n+r) = \frac{(n+p)!}{n!p!} \quad (2.10)$$

This traditional approach will be referred to as a “total-order expansion.”

An important alternative approach is to employ a “tensor-product expansion,” in which polynomial order bounds are applied on a per-dimension basis (no total-order bound is enforced) and all combinations of the one-dimensional polynomials are included. That is, the multi-index defining the set of  $\Psi_j$  is constrained by

$$m_i^j \leq p_i \quad (2.11)$$

where  $p_i$  is the polynomial order bound for the  $i$ -th dimension. In this case, the example basis for  $p = 2, n = 2$  is

$$\begin{aligned} \Psi_0(\boldsymbol{\xi}) &= \psi_0(\xi_1) \psi_0(\xi_2) = 1 \\ \Psi_1(\boldsymbol{\xi}) &= \psi_1(\xi_1) \psi_0(\xi_2) = \xi_1 \\ \Psi_2(\boldsymbol{\xi}) &= \psi_2(\xi_1) \psi_0(\xi_2) = \xi_1^2 - 1 \\ \Psi_3(\boldsymbol{\xi}) &= \psi_0(\xi_1) \psi_1(\xi_2) = \xi_2 \\ \Psi_4(\boldsymbol{\xi}) &= \psi_1(\xi_1) \psi_1(\xi_2) = \xi_1 \xi_2 \\ \Psi_5(\boldsymbol{\xi}) &= \psi_2(\xi_1) \psi_1(\xi_2) = (\xi_1^2 - 1) \xi_2 \\ \Psi_6(\boldsymbol{\xi}) &= \psi_0(\xi_1) \psi_2(\xi_2) = \xi_2^2 - 1 \\ \Psi_7(\boldsymbol{\xi}) &= \psi_1(\xi_1) \psi_2(\xi_2) = \xi_1 (\xi_2^2 - 1) \\ \Psi_8(\boldsymbol{\xi}) &= \psi_2(\xi_1) \psi_2(\xi_2) = (\xi_1^2 - 1) (\xi_2^2 - 1) \end{aligned}$$

and the total number of terms  $N_t$  is

$$N_t = 1 + P = \prod_{i=1}^n (p_i + 1) \quad (2.12)$$

It is apparent from Eq. 2.12 that the tensor-product expansion readily supports anisotropy in polynomial order for each dimension, since the polynomial order bounds for each dimension can be specified independently. It is also feasible to support anisotropy with total-order expansions, through pruning polynomials that satisfy the total-order bound but violate individual per-dimension bounds (the number of these pruned polynomials would then be subtracted from Eq. 2.10). Finally, custom tailoring of the expansion form can also be explored, e.g. to closely synchronize with monomial coverage in sparse grids through use of a summation of tensor expansions (see Section 2.7.3). In all cases, the specifics of the expansion are codified in the multi-index, and subsequent machinery for estimating response values and statistics from the expansion can be performed in a manner that is agnostic to the specific expansion form.

## 2.5 Stochastic Collocation

The SC expansion is formed as a sum of a set of multidimensional Lagrange interpolation polynomials, one polynomial per unique collocation point. Since these polynomials have the feature of being equal to 1 at their particular collocation point and 0 at all other points<sup>1</sup>, the coefficients of the expansion are just the response values at each of the collocation points. This can be written as:

$$R \cong \sum_{j=1}^{N_p} r_j L_j(\boldsymbol{\xi}) \quad (2.13)$$

where the set of  $N_p$  collocation points involves a structured multidimensional grid (a tensor-product grid as in Eq. 2.3 or a Smolyak sparse grid). There is no need for tailoring of the expansion form as there is for PCE (i.e., to

<sup>1</sup>for tensor interpolants and sparse interpolants based on fully nested rules (e.g., Clenshaw-Curtis, Gauss-Patterson, Genz-Keister); sparse interpolants based on non-nested rules will exhibit some interpolation error at the collocation points

synchronize the expansion polynomials with the set of integrable monomials) since the polynomials that appear in the expansion are determined by the Lagrange construction (Eq. 2.1). That is, any tailoring or refinement of the expansion occurs through the selection of points in the interpolation grid and the polynomial orders of the basis are adapted implicitly.

## 2.6 Transformations to uncorrelated standard variables

Polynomial chaos and stochastic collocation are expanded using polynomials that are functions of independent standard random variables  $\xi$ . Thus, a key component of either approach is performing a transformation of variables from the original random variables  $x$  to independent standard random variables  $\xi$  and then applying the stochastic expansion in the transformed space. This notion of independent standard space is extended over the notion of “u-space” used in reliability methods (see Section 1.1.2) in that it extends the standardized set beyond standard normals. For distributions that are already independent, three different approaches are of interest:

1. *Extended basis:* For each Askey distribution type, employ the corresponding Askey basis (Table 2.1). For non-Askey types, numerically generate an optimal polynomial basis for each independent distribution as described in Section 2.2. With usage of the optimal basis corresponding to each of the random variable types, we can exploit basis orthogonality under expectation (e.g., Eq. 2.16) without requiring a transformation of variables, thereby avoiding inducing additional nonlinearity that could slow convergence.
2. *Askey basis:* For non-Askey types, perform a nonlinear variable transformation from a given input distribution to the most similar Askey basis. For example, lognormal distributions might employ a Hermite basis in a transformed standard normal space and loguniform, triangular, and histogram distributions might employ a Legendre basis in a transformed standard uniform space. All distributions then employ the Askey orthogonal polynomials and their associated Gauss points/weights.
3. *Wiener basis:* For non-normal distributions, employ a nonlinear variable transformation to standard normal distributions. All distributions then employ the Hermite orthogonal polynomials and their associated Gauss points/weights.

For dependent distributions, we must first perform a nonlinear variable transformation to uncorrelated standard normal distributions, due to the independence of decorrelated standard normals. This involves the Nataf transformation, described in the following paragraph. We then have the following choices:

1. *Single transformation:* Following the Nataf transformation to independent standard normal distributions, employ the Wiener basis in the transformed space.
2. *Double transformation:* From independent standard normal space, transform back to either the original marginal distributions or the desired Askey marginal distributions and employ an extended or Askey basis, respectively, in the transformed space. Independence is maintained, but the nonlinearity of the Nataf transformation is at least partially mitigated.

DAKOTA currently supports single transformations for dependent variables in combination with an Askey basis for independent variables.

The transformation from correlated non-normal distributions to uncorrelated standard normal distributions is denoted as  $\xi = T(x)$  with the reverse transformation denoted as  $x = T^{-1}(\xi)$ . These transformations are nonlinear in general, and possible approaches include the Rosenblatt [59], Nataf [17], and Box-Cox [8] transformations. The

results in this paper employ the Nataf transformation, which is suitable for the common case when marginal distributions and a correlation matrix are provided, but full joint distributions are not known<sup>2</sup>. The Nataf transformation occurs in the following two steps. To transform between the original correlated x-space variables and correlated standard normals (“z-space”), a CDF matching condition is applied for each of the marginal distributions:

$$\Phi(z_i) = F(x_i) \quad (2.14)$$

where  $\Phi()$  is the standard normal cumulative distribution function and  $F()$  is the cumulative distribution function of the original probability distribution. Then, to transform between correlated z-space variables and uncorrelated  $\xi$ -space variables, the Cholesky factor  $\mathbf{L}$  of a modified correlation matrix is used:

$$\mathbf{z} = \mathbf{L}\boldsymbol{\xi} \quad (2.15)$$

where the original correlation matrix for non-normals in x-space has been modified to represent the corresponding “warped” correlation in z-space [17].

## 2.7 Spectral projection

The major practical difference between PCE and SC is that, in PCE, one must estimate the coefficients for known basis functions, whereas in SC, one must form the interpolants for known coefficients. PCE estimates its coefficients using either spectral projection or linear regression, where the former approach involves numerical integration based on random sampling, tensor-product quadrature, Smolyak sparse grids, or cubature methods. In SC, the multidimensional interpolants need to be formed over structured data sets, such as point sets from quadrature or sparse grids; approaches based on random sampling may not be used.

The spectral projection approach projects the response against each basis function using inner products and employs the polynomial orthogonality properties to extract each coefficient. Similar to a Galerkin projection, the residual error from the approximation is rendered orthogonal to the selected basis. From Eq. 2.8, taking the inner product of both sides with respect to  $\Psi_j$  and enforcing orthogonality yields:

$$\alpha_j = \frac{\langle R, \Psi_j \rangle}{\langle \Psi_j^2 \rangle} = \frac{1}{\langle \Psi_j^2 \rangle} \int_{\Omega} R \Psi_j \varrho(\boldsymbol{\xi}) d\boldsymbol{\xi}, \quad (2.16)$$

where each inner product involves a multidimensional integral over the support range of the weighting function. In particular,  $\Omega = \Omega_1 \otimes \cdots \otimes \Omega_n$ , with possibly unbounded intervals  $\Omega_j \subset \mathbb{R}$  and the tensor product form  $\varrho(\boldsymbol{\xi}) = \prod_{i=1}^n \varrho_i(\xi_i)$  of the joint probability density (weight) function. The denominator in Eq. 2.16 is the norm squared of the multivariate orthogonal polynomial, which can be computed analytically using the product of univariate norms squared

$$\langle \Psi_j^2 \rangle = \prod_{i=1}^n \langle \psi_{m_i}^2 \rangle \quad (2.17)$$

where the univariate inner products have simple closed form expressions for each polynomial in the Askey scheme [1] and are readily computed as part of the numerically-generated solution procedures described in Section 2.2. Thus, the primary computational effort resides in evaluating the numerator, which is evaluated numerically using sampling, quadrature, cubature, or sparse grid approaches (and this numerical approximation leads to use of the term “pseudo-spectral” by some investigators).

<sup>2</sup>If joint distributions are known, then the Rosenblatt transformation is preferred.

### 2.7.1 Sampling

In the sampling approach, the integral evaluation is equivalent to computing the expectation (mean) of the response-basis function product (the numerator in Eq. 2.16) for each term in the expansion when sampling within the density of the weighting function. This approach is only valid for PCE and since sampling does not provide any particular monomial coverage guarantee, it is common to combine this coefficient estimation approach with a total-order chaos expansion.

In computational practice, coefficient estimations based on sampling benefit from first estimating the response mean (the first PCE coefficient) and then removing the mean from the expectation evaluations for all subsequent coefficients. While this has no effect for quadrature/sparse grid methods (see following two sections) and little effect for fully-resolved sampling, it does have a small but noticeable beneficial effect for under-resolved sampling.

### 2.7.2 Tensor product quadrature

In quadrature-based approaches, the simplest general technique for approximating multidimensional integrals, as in Eq. 2.16, is to employ a tensor product of one-dimensional quadrature rules. Since there is little benefit to the use of nested quadrature rules in the tensor-product case<sup>3</sup>, we choose Gaussian abscissas, i.e. the zeros of polynomials that are orthogonal with respect to a density function weighting, e.g. Gauss-Hermite, Gauss-Legendre, Gauss-Laguerre, generalized Gauss-Laguerre, Gauss-Jacobi, or numerically-generated Gauss rules.

We first introduce an index  $i \in \mathbb{N}_+$ ,  $i \geq 1$ . Then, for each value of  $i$ , let  $\{\xi_1^i, \dots, \xi_{m_i}^i\} \subset \Omega_i$  be a sequence of abscissas for quadrature on  $\Omega_i$ . For  $f \in C^0(\Omega_i)$  and  $n = 1$  we introduce a sequence of one-dimensional quadrature operators

$$\mathcal{W}^i(f)(\xi) = \sum_{j=1}^{m_i} f(\xi_j^i) w_j^i, \quad (2.18)$$

with  $m_i \in \mathbb{N}$  given. When utilizing Gaussian quadrature, Eq. 2.18 integrates exactly all polynomials of degree less than  $2m_i - 1$ , for each  $i = 1, \dots, n$ . Given an expansion order  $p$ , the highest order coefficient evaluations (Eq. 2.16) can be assumed to involve integrands of at least polynomial order  $2p$  ( $\Psi$  of order  $p$  and  $R$  modeled to order  $p$ ) in each dimension such that a minimal Gaussian quadrature order of  $p + 1$  will be required to obtain good accuracy in these coefficients.

Now, in the multivariate case  $n > 1$ , for each  $f \in C^0(\Omega)$  and the multi-index  $\mathbf{i} = (i_1, \dots, i_n) \in \mathbb{N}_+^n$  we define the full tensor product quadrature formulas

$$\mathcal{Q}_{\mathbf{i}}^n f(\xi) = (\mathcal{W}^{i_1} \otimes \dots \otimes \mathcal{W}^{i_n})(f)(\xi) = \sum_{j_1=1}^{m_{i_1}} \dots \sum_{j_n=1}^{m_{i_n}} f(\xi_{j_1}^{i_1}, \dots, \xi_{j_n}^{i_n}) (w_{j_1}^{i_1} \otimes \dots \otimes w_{j_n}^{i_n}). \quad (2.19)$$

Clearly, the above product needs  $\prod_{j=1}^n m_{i_j}$  function evaluations. Therefore, when the number of input random variables is small, full tensor product quadrature is a very effective numerical tool. On the other hand, approximations based on tensor product grids suffer from the *curse of dimensionality* since the number of collocation points in a tensor grid grows exponentially fast in the number of input random variables. For example, if Eq. 2.19 employs the same order for all random dimensions,  $m_{i_j} = m$ , then Eq. 2.19 requires  $m^n$  function evaluations.

In [22], it is demonstrated that close synchronization of expansion form with the monomial resolution of a particular numerical integration technique can result in significant performance improvements. In particular, the traditional approach of employing a total-order PCE (Eqs. 2.9–2.10) neglects a significant portion of the monomial coverage for a tensor-product quadrature approach, and one should rather employ a tensor-product PCE

<sup>3</sup>Unless a refinement procedure is in use.

(Eqs. 2.11–2.12) to provide improved synchronization and more effective usage of the Gauss point evaluations. When the quadrature points are standard Gauss rules (i.e., no Clenshaw-Curtis, Gauss-Patterson, or Genz-Keister nested rules), it has been shown that tensor-product PCE and SC result in identical polynomial forms [15], completely eliminating a performance gap that exists between total-order PCE and SC [22].

### 2.7.3 Smolyak sparse grids

If the number of random variables is moderately large, one should rather consider sparse tensor product spaces as first proposed by Smolyak [62] and further investigated by Refs. [32, 6, 29, 77, 48, 49] that reduce dramatically the number of collocation points, while preserving a high level of accuracy.

Here we follow the notation and extend the description in Ref. [48] to describe the Smolyak *isotropic* formulas  $\mathcal{A}(w, n)$ , where  $w$  is a level that is independent of dimension<sup>4</sup>. The Smolyak formulas are just linear combinations of the product formulas in Eq. 2.19 with the following key property: only products with a relatively small number of points are used. With  $\mathcal{U}^0 = 0$  and for  $i \geq 1$  define

$$\Delta^i = \mathcal{U}^i - \mathcal{U}^{i-1}. \quad (2.20)$$

and we set  $|\mathbf{i}| = i_1 + \dots + i_n$ . Then the isotropic Smolyak quadrature formula is given by

$$\mathcal{A}(w, n) = \sum_{|\mathbf{i}| \leq w+n} (\Delta^{i_1} \otimes \dots \otimes \Delta^{i_n}). \quad (2.21)$$

Equivalently, formula Eq. 2.21 can be written as [69]

$$\mathcal{A}(w, n) = \sum_{w+1 \leq |\mathbf{i}| \leq w+n} (-1)^{w+n-|\mathbf{i}|} \binom{n-1}{w+n-|\mathbf{i}|} \cdot (\mathcal{U}^{i_1} \otimes \dots \otimes \mathcal{U}^{i_n}). \quad (2.22)$$

For each index set  $\mathbf{i}$  of levels, linear or nonlinear growth rules are used to define the corresponding one-dimensional quadrature orders. The following growth rules are employed for indices  $i \geq 1$ , where closed and open refer to the inclusion and exclusion of the bounds within an interval, respectively:

$$\text{closed nonlinear : } m = \begin{cases} 1 & i = 1 \\ 2^{i-1} + 1 & i > 1 \end{cases} \quad (2.23)$$

$$\text{open nonlinear : } m = 2^i - 1 \quad (2.24)$$

$$\text{open linear : } m = 2i - 1 \quad (2.25)$$

Nonlinear growth rules are used for fully nested rules (e.g., Clenshaw-Curtis is closed fully nested and Gauss-Patterson is open fully nested), and linear growth rules are best for standard Gauss rules that take advantage of, at most, “weak” nesting (e.g., reuse of the center point).

Examples of isotropic sparse grids, constructed from the fully nested Clenshaw-Curtis abscissas and the weakly-nested Gaussian abscissas are shown in Figure 2.1, where  $\Omega = [-1, 1]^2$  and both Clenshaw-Curtis and Gauss-Legendre employ nonlinear growth<sup>5</sup> from Eqs. 2.23 and 2.24, respectively. There, we consider a two-dimensional parameter space and a maximum level  $w = 5$  (sparse grid  $\mathcal{A}(5, 2)$ ). To see the reduction in function evaluations with respect to full tensor product grids, we also include a plot of the corresponding Clenshaw-Curtis isotropic full tensor grid having the same maximum number of points in each direction, namely  $2^w + 1 = 33$ .

<sup>4</sup>Other common formulations use a dimension-dependent level  $q$  where  $q \geq n$ . We use  $w = q - n$ , where  $w \geq 0$  for all  $n$ .

<sup>5</sup>We prefer linear growth for Gauss-Legendre, but employ nonlinear growth here for purposes of comparison.



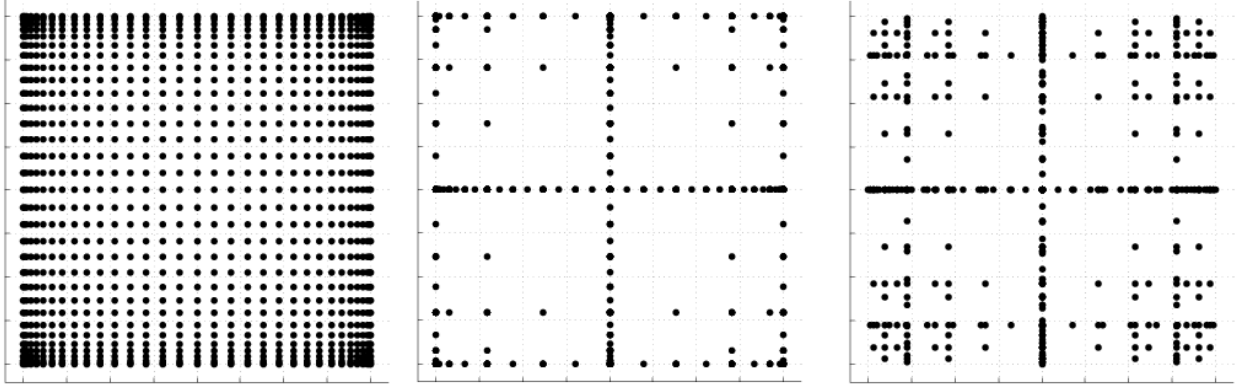


Figure 2.1: Two-dimensional grid comparison with a tensor product grid using Clenshaw-Curtis points (left) and sparse grids  $\mathcal{A}(5, 2)$  utilizing Clenshaw-Curtis (middle) and Gauss-Legendre (right) points with nonlinear growth.

In [22], it is demonstrated that the synchronization of total-order PCE with the monomial resolution of a sparse grid is imperfect, and that sparse grid SC consistently outperforms sparse grid PCE when employing the sparse grid to directly evaluate the integrals in Eq. 2.16. In our DAKOTA implementation, we depart from the use of sparse integration of total-order expansions, and instead employ a linear combination of tensor expansions [14]. That is, we compute separate tensor polynomial chaos expansions for each of the underlying tensor quadrature grids (for which there is no synchronization issue) and then sum them using the Smolyak combinatorial coefficient (from Eq. 2.22 in the isotropic case). This improves accuracy, preserves the PCE/SC consistency property described in Section 2.7.2, and also simplifies PCE for the case of anisotropic sparse grids described next.

For anisotropic Smolyak sparse grids, a dimension preference vector is used to emphasize important stochastic dimensions. Given a mechanism for defining anisotropy, we can extend the definition of the sparse grid from that of Eq. 2.22 to weight the contributions of different index set components. First, the sparse grid index set constraint becomes

$$w\underline{\gamma} < \mathbf{i} \cdot \underline{\gamma} \leq w\underline{\gamma} + |\underline{\gamma}| \quad (2.26)$$

where  $\underline{\gamma}$  is the minimum of the dimension weights  $\gamma_k$ ,  $k = 1$  to  $n$ . The dimension weighting vector  $\underline{\gamma}$  amplifies the contribution of a particular dimension index within the constraint, and is therefore inversely related to the dimension preference (higher weighting produces lower index set levels). For the isotropic case of all  $\gamma_k = 1$ , it is evident that you reproduce the isotropic index constraint  $w + 1 \leq |\mathbf{i}| \leq w + n$  (note the change from  $<$  to  $\leq$ ). Second, the combinatorial coefficient for adding the contribution from each of these index sets is modified as described in [10].

### 2.7.4 Cubature

Cubature rules [63, 76] are specifically optimized for multidimensional integration and are distinct from tensor-products and sparse grids in that they are not based on combinations of one-dimensional Gauss quadrature rules. They have the advantage of improved scalability to large numbers of random variables, but are restricted in integrand order and require homogeneous random variable sets (achieved via transformation). For example, optimal rules for integrands of 2, 3, and 5 and either Gaussian or uniform densities allow low-order polynomial chaos expansions ( $p = 1$  or 2) that are useful for global sensitivity analysis including main effects and, for  $p = 2$ , all two-way interactions.

## 2.8 Linear regression

The linear regression approach uses a single linear least squares solution of the form:

$$\Psi \alpha = R \quad (2.27)$$

to solve for the complete set of PCE coefficients  $\alpha$  that best match a set of response values  $R$ . The set of response values is obtained either by performing a design of computer experiments within the density function of  $\xi$  (point collocation [68, 42]) or from a subset of tensor quadrature points with highest product weight (probabilistic collocation [65]). In either case, each row of the matrix  $\Psi$  contains the  $N_t$  multivariate polynomial terms  $\Psi_j$  evaluated at a particular  $\xi$  sample. An over-sampling is recommended in the case of random samples ([42] recommends  $2N_t$  samples), resulting in a least squares solution for the over-determined system. As for sampling-based coefficient estimation, this approach is only valid for PCE and does not require synchronization with monomial coverage; thus it is common to combine this coefficient estimation approach with a traditional total-order chaos expansion in order to keep sampling requirements low. In this case, simulation requirements for this approach scale as  $\frac{r(n+p)!}{n!p!}$  ( $r$  is an over-sampling factor with typical values  $1 \leq r \leq 2$ ), which can be significantly more affordable than isotropic tensor-product quadrature (scales as  $(p+1)^n$  for standard Gauss rules) for larger problems. Finally, additional regression equations can be obtained through the use of derivative information (gradients and Hessians) from each collocation point, which can aid in scaling with respect to the number of random variables, particularly for adjoint-based derivative approaches.

## 2.9 Analytic moments

Mean and covariance of polynomial chaos expansions are available in simple closed form:

$$\mu_i = \langle R_i \rangle \cong \sum_{k=0}^P \alpha_{ik} \langle \Psi_k(\xi) \rangle = \alpha_{i0} \quad (2.28)$$

$$\Sigma_{ij} = \langle (R_i - \mu_i)(R_j - \mu_j) \rangle \cong \sum_{k=1}^P \sum_{l=1}^P \alpha_{ik} \alpha_{jl} \langle \Psi_k(\xi) \Psi_l(\xi) \rangle = \sum_{k=1}^P \alpha_{ik} \alpha_{jk} \langle \Psi_k^2 \rangle \quad (2.29)$$

where the norm squared of each multivariate polynomial is computed from Eq. 2.17. These expressions provide exact moments of the expansions, which converge under refinement to moments of the true response functions.

Similar expressions can be derived for stochastic collocation:

$$\mu_i = \langle R_i \rangle \cong \sum_{k=1}^{N_p} r_{ik} \langle L_k(\xi) \rangle = \sum_{k=1}^{N_p} r_{ik} w_k \quad (2.30)$$

$$\Sigma_{ij} = \langle R_i R_j \rangle - \mu_i \mu_j \cong \sum_{k=1}^{N_p} \sum_{l=1}^{N_p} r_{ik} r_{jl} \langle L_k(\xi) L_l(\xi) \rangle - \mu_i \mu_j = \sum_{k=1}^{N_p} r_{ik} r_{jk} w_k - \mu_i \mu_j \quad (2.31)$$

where we have simplified the expectation of Lagrange polynomials constructed at Gauss points and then integrated at these same Gauss points. For tensor grids and sparse grids with fully nested rules, these expectations leave only the weight corresponding to the point for which the interpolation value is one, such that the final equalities in Eqs. 2.30–2.31 hold precisely. For sparse grids with non-nested rules, however, interpolation error exists at the collocation points, such that these final equalities hold only approximately. In this case, we have the choice of computing the moments based on sparse numerical integration or based on the moments of the (imperfect)

sparse interpolant, where small differences may exist prior to numerical convergence. In DAKOTA, we employ the former approach; i.e., the right-most expressions in Eqs. 2.30–2.31 are employed for all tensor and sparse cases irregardless of nesting. Skewness and kurtosis calculations as well as sensitivity derivations in the following sections are also based on this choice. The expressions for skewness and (excess) kurtosis from direct numerical integration of the response function are as follows:

$$\gamma_{1_i} = \left\langle \left( \frac{R_i - \mu_i}{\sigma_i} \right)^3 \right\rangle \cong \frac{1}{\sigma_i^3} \left[ \sum_{k=1}^{N_p} (r_{ik} - \mu_i)^3 w_k \right] \quad (2.32)$$

$$\gamma_{2_i} = \left\langle \left( \frac{R_i - \mu_i}{\sigma_i} \right)^4 \right\rangle - 3 \cong \frac{1}{\sigma_i^4} \left[ \sum_{k=1}^{N_p} (r_{ik} - \mu_i)^4 w_k \right] - 3 \quad (2.33)$$

## 2.10 Local sensitivity analysis: derivatives with respect to expansion variables

Polynomial chaos expansions are easily differentiated with respect to the random variables [57]. First, using Eq. 2.8,

$$\frac{dR}{d\xi_i} = \sum_{j=0}^P \alpha_j \frac{d\Psi_j(\xi)}{d\xi_i} \quad (2.34)$$

and then using Eq. 2.7,

$$\frac{d\Psi_j(\xi)}{d\xi_i} = \frac{d\psi_i}{d\xi_i} \prod_{\substack{k=1 \\ k \neq i}}^n \psi_{m_k^j}(\xi_k) \quad (2.35)$$

where the univariate polynomial derivatives  $\frac{d\psi_i}{d\xi_i}$  have simple closed form expressions for each polynomial in the Askey scheme [1]. Finally, using the Jacobian of the (extended) Nataf variable transformation,

$$\frac{dR}{dx_i} = \frac{dR}{d\xi_i} \frac{d\xi_i}{dx_i} \quad (2.36)$$

which simplifies to  $\frac{dR}{d\xi_i} \frac{d\xi_i}{dx_i}$  in the case of uncorrelated  $x_i$ .

Similar expressions may be derived for stochastic collocation, starting from Eq. 2.13:

$$\frac{dR}{d\xi_i} = \sum_{j=1}^{N_p} r_j \frac{dL_j(\xi)}{d\xi_i} \quad (2.37)$$

where the multidimensional interpolant  $L_j$  is formed over either tensor-product quadrature points or a Smolyak sparse grid. For the former case, the derivative of the multidimensional interpolant  $L_j$  involves a product rule of the one-dimensional interpolants  $L_k$ :

$$\frac{dL_j(\xi)}{d\xi_i} = \frac{dL_i}{d\xi_i} \prod_{\substack{k=1 \\ k \neq i}}^n L_k(\xi_k) \quad (2.38)$$

and for the latter case, the derivative involves a linear combination of these product rules, as dictated by the Smolyak recursion shown in Eq. 2.22. Finally, calculation of  $\frac{dR}{dx_i}$  involves the same Jacobian application shown in Eq. 2.36.

## 2.11 Global sensitivity analysis: variance-based decomposition

In addition to obtaining derivatives of stochastic expansions with respect to the random variables, it is possible to obtain variance-based sensitivity indices from the stochastic expansions. Variance-based sensitivity indices are explained in the Design of Experiments Chapter of the Users Manual. The concepts are summarized here as well. Variance-based decomposition is a global sensitivity method that summarizes how the uncertainty in model output can be apportioned to uncertainty in individual input variables. VBD uses two primary measures, the main effect sensitivity index  $S_i$  and the total effect index  $T_i$ . These indices are also called the Sobol' indices. The main effect sensitivity index corresponds to the fraction of the uncertainty in the output,  $Y$ , that can be attributed to input  $x_i$  alone. The total effects index corresponds to the fraction of the uncertainty in the output,  $Y$ , that can be attributed to input  $x_i$  and its interactions with other variables. The main effect sensitivity index compares the variance of the conditional expectation  $Var_{x_i}[E(Y|x_i)]$  against the total variance  $Var(Y)$ . Formulas for the indices are:

$$S_i = \frac{Var_{x_i}[E(Y|x_i)]}{Var(Y)} \quad (2.39)$$

and

$$T_i = \frac{E(Var(Y|x_{-i}))}{Var(Y)} = \frac{Var(Y) - Var(E[Y|x_{-i}])}{Var(Y)} \quad (2.40)$$

where  $Y = f(\mathbf{x})$  and  $x_{-i} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_m)$ .

The calculation of  $S_i$  and  $T_i$  requires the evaluation of m-dimensional integrals which are typically approximated by Monte-Carlo sampling. However, in stochastic expansion methods, it is possible to obtain the sensitivity indices as analytic functions of the coefficients in the stochastic expansion. The derivation of these results is presented in [64]. The sensitivity indices are printed as a default when running either polynomial chaos or stochastic collocation in DAKOTA. Note that in addition to the first-order main effects,  $S_i$ , we are able to calculate the sensitivity indices for higher order interactions such as the two-way interaction  $S_{i,j}$ .

## 2.12 Automated Refinement

Several approaches for refinement of stochastic expansions are presented here: uniform p-refinement with isotropic sparse and tensor grids, adaptive p-refinement using anisotropic sparse and tensor grids, and goal-oriented adaptive p-refinement using generalized sparse grids. Each involves incrementing the grid upon which the stochastic expansions are based, using differing refinement criteria and convergence controls.

### 2.12.1 Uniform p-refinement with isotropic grids

Uniform p-refinement involves ramping the order of a tensor-product quadrature grid or the level of a Smolyak sparse grid isotropically. In this case, dimension preferences are not computed, and the only algorithmic requirements are:

- With the usage of nested integration rules with restricted exponential growth, DAKOTA must ensure that a change in level results in a sufficient change in the grid; otherwise premature convergence could occur within the refinement process. If no change is initially detected, DAKOTA continues incrementing the order/level (without grid evaluation) until the number of grid points increases.

- A convergence criterion is required. For uniform refinement, DAKOTA employs the  $L^2$  norm of the change in the response covariance matrix as a general-purpose convergence metric.

### 2.12.2 Adaptive p-refinement with anisotropic grids

Adaptive p-refinement involves ramping the order of a tensor-product quadrature grid or the level of a Smolyak sparse grid anisotropically, that is, using a defined dimension preference. This dimension preference may be computed from local sensitivity analysis, global sensitivity analysis, a posteriori error estimation, or decay rate estimation. In the current release, we focus on global sensitivity analysis from low order isotropic expansions where dimension preference is defined from total Sobol' indices (Eq. 2.40) and is updated on every iteration. This dimension preference vector supports anisotropic sparse grids based on a linear index-set constraint (Eq. 2.26) or anisotropic tensor grids (Eq. 2.19) with dimension order scaled proportionately to preference; in both cases, dimension refinement lower bound constraints are enforced to ensure that all previously evaluated points remain in new refined grids.

### 2.12.3 Goal-oriented p-refinement with generalized sparse grids

The uniform and adaptive refinement capabilities described above define anisotropy and control convergence in a highly structured manner based on variance-related measures. The generalized sparse grid algorithm [33], on the other hand, supports greater flexibility in the definition of sparse grid index sets and supports refinement controls based on general statistical quantities of interest (QOI). This algorithm was originally intended for adaptive numerical integration on a hypercube, but can be readily extended to the adaptive refinement of stochastic expansions using the following customizations:

- In addition to hierarchical interpolants in SC, we employ independent polynomial chaos expansions for each active and accepted index set. Pushing and popping index sets then involves increments of tensor chaos expansions (as described in Section 2.7.3) along with corresponding increments to the Smolyak combinatorial coefficients.
- Since we support bases for more than uniform distributions on a hypercube, we exploit rule nesting when possible (i.e., Gauss-Patterson for uniform or transformed uniform variables, and Genz-Keister for normal or transformed normal variables), but we do not require it. This implies a loss of some algorithmic simplifications in [33] that occur when grids are strictly hierarchical.
- In the evaluation of the effect of a trial index set, the goal in [33] is numerical integration and the metric is the size of the increment induced by the trial set on the expectation of the function of interest. It is straightforward to instead measure the effect of a trial index set on response covariance, numerical probability, or other statistical QOI computed by post-processing the resulting PCE or SC expansion. By tying the refinement process more closely to the statistical QOI, the refinement process can become more efficient in achieving the desired analysis objectives.

Given these customizations, the algorithmic steps can be summarized as:

1. *Initialization:* Starting from an initial isotropic or anisotropic reference grid (often the  $w = 0$  grid corresponding to a single collocation point), accept the reference index sets as the old set and define active index sets using the admissible forward neighbors of all old index sets.
2. *Trial set evaluation:* Evaluate the tensor grid corresponding to each trial active index set, form the tensor polynomial chaos expansion or tensor interpolant corresponding to it, update the Smolyak combinatorial

coefficients, and combine the trial expansion with the reference expansion. Perform necessary bookkeeping to allow efficient restoration of previously evaluated tensor expansions.

3. *Trial set selection:* Select the trial index set that induces the largest change in the statistical quantity of interest. In our implementation, we employ an  $L^2$  norm of change in CDF/CCDF probability/reliability/response level mappings, when level mappings are present, or  $L^2$  norm of change in response covariance, when level mappings are not present.
4. *Update sets:* If the largest change induced by the trial sets exceeds a specified convergence tolerance, then promote the selected trial set from the active set to the old set and update the active sets with new admissible forward neighbors; return to step 2 and evaluate all trial sets with respect to the new reference point. If the convergence tolerance is satisfied, advance to step 5.
5. *Finalization:* Promote all remaining active sets to the old set, update the Smolyak combinatorial coefficients, and perform a final combination of tensor expansions to arrive at the final result for the statistical quantity of interest.

## Chapter 3

# Epistemic Methods

This chapter covers theoretical aspects of methods for propagating epistemic uncertainty.

### 3.1 Dempster-Shafer theory of evidence (DSTE)

In Dempster-Shafer theory, the event space is defined by a triple  $(\mathcal{S}, \mathbb{S}, m)$  which defines  $\mathcal{S}$  the universal set,  $\mathbb{S}$  a countable collection of subsets of  $\mathcal{S}$ , and a notional measure  $m$ .  $\mathcal{S}$  and  $\mathbb{S}$  have a similar meaning to that in classical probability theory; the main difference is that  $\mathbb{S}$ , also known as the focal elements, does not have to be a  $\sigma$ -algebra over  $\mathcal{S}$ . The operator  $m$  is defined to be

$$m(\mathcal{U}) = \begin{cases} > 0 & \text{if } \mathcal{U} \in \mathbb{S} \\ 0 & \text{if } \mathcal{U} \subset \mathcal{S} \text{ and } \mathcal{U} \notin \mathbb{S} \end{cases} \quad (3.1)$$

$$\sum_{\mathcal{U} \in \mathbb{S}} m(\mathcal{U}) = 1 \quad (3.2)$$

where  $m(\mathcal{U})$  is known as the basic probability assignment (BPA) of the set  $\mathcal{U}$ . In the DSTE framework, belief and plausibility are defined as:

$$\text{Bel}(\mathcal{E}) = \sum_{\{\mathcal{U} \mid \mathcal{U} \subset \mathcal{E}, \mathcal{U} \in \mathbb{S}\}} m(\mathcal{U}) \quad (3.3)$$

$$\text{Pl}(\mathcal{E}) = \sum_{\{\mathcal{U} \mid \mathcal{U} \cap \mathcal{E} \neq \emptyset, \mathcal{U} \in \mathbb{S}\}} m(\mathcal{U}) \quad (3.4)$$

The belief  $\text{Bel}(\mathcal{E})$  is interpreted to be the minimum likelihood that is associated with the event  $\mathcal{E}$ . Similarly, the plausibility  $\text{Pl}(\mathcal{E})$  is the maximum amount of likelihood that could be associated with  $\mathcal{E}$ . This particular structure allows us to handle unconventional inputs, such as conflicting pieces of evidence (e.g. dissenting expert opinions), that would be otherwise discarded in an interval analysis or probabilistic framework. The ability to make use of this information results in a commensurately more informed output.

The procedure to compute belief structures involves four major steps:

1. Determine the set of  $d$ -dimensional hypercubes that have a nonzero evidential measure
2. Compute the composite evidential measure (BPA) of each hypercube
3. Propagate each hypercube through the model and obtain the response bounds within each hypercube
4. Aggregate the minimum and maximum values of the response per hypercube with the BPAs to obtain cumulative belief and plausibility functions on the response (e.g. calculate a belief structure on the response).

The first step involves identifying combinations of focal elements defined on the inputs that define a hypercube. The second step involves defining an aggregate BPA for that hypercube, which is the product of the BPAs of the individual focal elements defining the hypercube. The third step involves finding the maximum and minimum values of the response value in each hypercube, and this part can be very computationally expensive. Finally, the results over all hypercubes are aggregated to form belief structures on the response.



## Chapter 4

# Surrogate-Based Local Minimization

A generally-constrained nonlinear programming problem takes the form

$$\begin{aligned}
 & \text{minimize} && f(\mathbf{x}) \\
 & \text{subject to} && \mathbf{g}_l \leq \mathbf{g}(\mathbf{x}) \leq \mathbf{g}_u \\
 & && \mathbf{h}(\mathbf{x}) = \mathbf{h}_t \\
 & && \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u
 \end{aligned} \tag{4.1}$$

where  $\mathbf{x} \in \mathbb{R}^n$  is the vector of design variables, and  $f$ ,  $\mathbf{g}$ , and  $\mathbf{h}$  are the objective function, nonlinear inequality constraints, and nonlinear equality constraints, respectively<sup>1</sup>. Individual nonlinear inequality and equality constraints are enumerated using  $i$  and  $j$ , respectively (e.g.,  $g_i$  and  $h_j$ ). The corresponding surrogate-based optimization (SBO) algorithm may be formulated in several ways and applied to either optimization or least-squares calibration problems. In all cases, SBO solves a sequence of  $k$  approximate optimization subproblems subject to a trust region constraint  $\Delta^k$ ; however, many different forms of the surrogate objectives and constraints in the approximate subproblem can be explored. In particular, the subproblem objective may be a surrogate of the original objective or a surrogate of a merit function (most commonly, the Lagrangian or augmented Lagrangian), and the subproblem constraints may be surrogates of the original constraints, linearized approximations of the surrogate constraints, or may be omitted entirely. Each of these combinations is shown in Table 4.1, where black indicates an inappropriate combination, gray indicates an acceptable combination, and blue indicates a common combination.

Initial approaches to nonlinearly-constrained SBO optimized an approximate merit function which incorporated the nonlinear constraints [58, 3]:

$$\begin{aligned}
 & \text{minimize} && \hat{\Phi}^k(\mathbf{x}) \\
 & \text{subject to} && \|\mathbf{x} - \mathbf{x}_c^k\|_\infty \leq \Delta^k
 \end{aligned} \tag{4.2}$$

<sup>1</sup>Any linear constraints are not approximated and may be added without modification to all formulations

Table 4.1: SBO approximate subproblem formulations.

	Original Objective	Lagrangian	Augmented Lagrangian
No constraints			TRAL
Linearized constraints		SQP-like	
Original constraints	Direct surrogate		IPTRSAO

where the surrogate merit function is denoted as  $\hat{\Phi}(\mathbf{x})$ ,  $\mathbf{x}_c$  is the center point of the trust region, and the trust region is truncated at the global variable bounds as needed. The merit function to approximate was typically chosen to be a standard implementation [67, 50, 35] of the augmented Lagrangian merit function (see Eqs. 4.11–4.12), where the surrogate augmented Lagrangian is constructed from individual surrogate models of the objective and constraints (approximate and assemble, rather than assemble and approximate). In Table 4.1, this corresponds to row 1, column 3, and is known as the trust-region augmented Lagrangian (TRAL) approach. While this approach was provably convergent, convergence rates to constrained minima have been observed to be slowed by the required updating of Lagrange multipliers and penalty parameters [53]. Prior to converging these parameters, SBO iterates did not strictly respect constraint boundaries and were often infeasible. A subsequent approach (IPTRSAO [53]) that sought to directly address this shortcoming added explicit surrogate constraints (row 3, column 3 in Table 4.1):

$$\begin{aligned} & \text{minimize} && \hat{\Phi}^k(\mathbf{x}) \\ & \text{subject to} && \mathbf{g}_l \leq \hat{\mathbf{g}}^k(\mathbf{x}) \leq \mathbf{g}_u \\ & && \hat{\mathbf{h}}^k(\mathbf{x}) = \mathbf{h}_t \\ & && \|\mathbf{x} - \mathbf{x}_c^k\|_\infty \leq \Delta^k. \end{aligned} \quad (4.3)$$

While this approach does address infeasible iterates, it still shares the feature that the surrogate merit function may reflect inaccurate relative weightings of the objective and constraints prior to convergence of the Lagrange multipliers and penalty parameters. That is, one may benefit from more feasible intermediate iterates, but the process may still be slow to converge to optimality. The concept of this approach is similar to that of SQP-like SBO approaches [3] which use linearized constraints:

$$\begin{aligned} & \text{minimize} && \hat{\Phi}^k(\mathbf{x}) \\ & \text{subject to} && \mathbf{g}_l \leq \hat{\mathbf{g}}^k(\mathbf{x}_c^k) + \nabla \hat{\mathbf{g}}^k(\mathbf{x}_c^k)^T (\mathbf{x} - \mathbf{x}_c^k) \leq \mathbf{g}_u \\ & && \hat{\mathbf{h}}^k(\mathbf{x}_c^k) + \nabla \hat{\mathbf{h}}^k(\mathbf{x}_c^k)^T (\mathbf{x} - \mathbf{x}_c^k) = \mathbf{h}_t \\ & && \|\mathbf{x} - \mathbf{x}_c^k\|_\infty \leq \Delta^k. \end{aligned} \quad (4.4)$$

in that the primary concern is minimizing a composite merit function of the objective and constraints, but under the restriction that the original problem constraints may not be wildly violated prior to convergence of Lagrange multiplier estimates. Here, the merit function selection of the Lagrangian function (row 2, column 2 in Table 4.1; see also Eq. 4.10) is most closely related to SQP, which includes the use of first-order Lagrange multiplier updates (Eq. 4.16) that should converge more rapidly near a constrained minimizer than the zeroth-order updates (Eqs. 4.13–4.14) used for the augmented Lagrangian.

All of these previous constrained SBO approaches involve a recasting of the approximate subproblem objective and constraints as a function of the original objective and constraint surrogates. A more direct approach is to use a formulation of:

$$\begin{aligned} & \text{minimize} && \hat{f}^k(\mathbf{x}) \\ & \text{subject to} && \mathbf{g}_l \leq \hat{\mathbf{g}}^k(\mathbf{x}) \leq \mathbf{g}_u \\ & && \hat{\mathbf{h}}^k(\mathbf{x}) = \mathbf{h}_t \\ & && \|\mathbf{x} - \mathbf{x}_c^k\|_\infty \leq \Delta^k \end{aligned} \quad (4.5)$$

This approach has been termed the direct surrogate approach since it optimizes surrogates of the original objective and constraints (row 3, column 1 in Table 4.1) without any recasting. It is attractive both from its simplicity and potential for improved performance, and is the default approach taken in DAKOTA. Other DAKOTA defaults include the use of a filter method for iterate acceptance (see Section 4.1), an augmented Lagrangian merit function (see Section 4.2), Lagrangian hard convergence assessment (see Section 4.3), and no constraint relaxation (see Section 4.4).

Table 4.2: Sample trust region ratio logic.

Ratio Value	Surrogate Accuracy	Iterate Acceptance	Trust Region Sizing
$\rho^k \leq 0$	poor	reject step	shrink
$0 < \rho^k \leq 0.25$	marginal	accept step	shrink
$0.25 < \rho^k < 0.75$ or $\rho^k > 1.25$	moderate	accept step	retain
$0.75 \leq \rho^k \leq 1.25$	good	accept step	expand <sup>2</sup>

While the formulation of Eq. 4.2 (and others from row 1 in Table 4.1) can suffer from infeasible intermediate iterates and slow convergence to constrained minima, each of the approximate subproblem formulations with explicit constraints (Eqs. 4.3-4.5, and others from rows 2-3 in Table 4.1) can suffer from the lack of a feasible solution within the current trust region. Techniques for dealing with this latter challenge involve some form of constraint relaxation. Homotopy approaches [53, 52] or composite step approaches such as Byrd-Omojokun [51], Celis-Dennis-Tapia [11], or MAESTRO [3] may be used for this purpose (see Section 4.4).

After each of the  $k$  iterations in the SBO method, the predicted step is validated by computing  $f(\mathbf{x}_*^k)$ ,  $\mathbf{g}(\mathbf{x}_*^k)$ , and  $\mathbf{h}(\mathbf{x}_*^k)$ . One approach forms the trust region ratio  $\rho^k$  which measures the ratio of the actual improvement to the improvement predicted by optimization on the surrogate model. When optimizing on an approximate merit function (Eqs. 4.2-4.4), the following ratio is natural to compute

$$\rho^k = \frac{\Phi(\mathbf{x}_c^k) - \Phi(\mathbf{x}_*^k)}{\hat{\Phi}(\mathbf{x}_c^k) - \hat{\Phi}(\mathbf{x}_*^k)}. \quad (4.6)$$

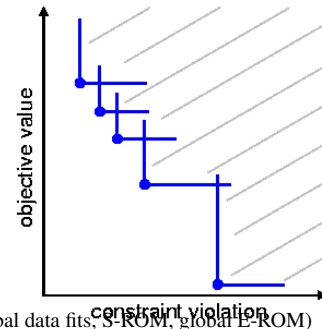
The formulation in Eq. 4.5 may also form a merit function for computing the trust region ratio; however, the omission of this merit function from explicit use in the approximate optimization cycles can lead to synchronization problems with the optimizer.

Once computed, the value for  $\rho^k$  can be used to define the step acceptance and the next trust region size  $\Delta^{k+1}$  using logic similar to that shown in Table 4.2. Typical factors for shrinking and expanding are 0.5 and 2.0, respectively, but these as well as the threshold ratio values are tunable parameters in the algorithm (see Surrogate-Based Method controls in the DAKOTA Reference Manual [2]). In addition, the use of discrete thresholds is not required, and continuous relationships using adaptive logic can also be explored [74, 75]. Iterate acceptance or rejection completes an SBO cycle, and the cycles are continued until either soft or hard convergence criteria (see Section 4.3) are satisfied.

## 4.1 Iterate acceptance logic

When a surrogate optimization is completed and the approximate solution has been validated, then the decision must be made to either accept or reject the step. The traditional approach is to base this decision on the value of the trust region ratio, as outlined previously in Table 4.2. An alternate approach is to utilize a filter method [28], which does not require penalty parameters or Lagrange multiplier estimates. The basic idea in a filter method is to apply the concept of Pareto optimality to the objective function and constraint violations and only accept an iterate if it is not dominated by any previous iterate. Mathematically, a new iterate is not dominated if at least one of the following:

$$\text{either } f < f^{(i)} \text{ or } c < c^{(i)} \quad (4.7)$$



<sup>2</sup>Exception: retain if  $\mathbf{x}_*^k$  in trust region interior for design of experiments-based surrogates (global data fits, S-ROM, global E-ROM)

is true for all  $i$  in the filter, where  $c$  is a selected norm of the constraint violation. This basic description can be augmented with mild requirements to prevent point accumulation and assure convergence, known as a slanting filter [28]. Figure 4.1 illustrates the filter concept, where objective values are plotted against constraint violation for accepted iterates (blue circles) to define the dominated region (denoted by the gray lines). A filter method relaxes the common enforcement of monotonicity in constraint violation reduction and, by allowing more flexibility in acceptable step generation, often allows the algorithm to be more efficient.

The use of a filter method is compatible with any of the SBO formulations in Eqs. 4.2–4.5.

## 4.2 Merit functions

The merit function  $\Phi(\mathbf{x})$  used in Eqs. 4.2-4.4,4.6 may be selected to be a penalty function, an adaptive penalty function, a Lagrangian function, or an augmented Lagrangian function. In each of these cases, the more flexible inequality and equality constraint formulations with two-sided bounds and targets (Eqs. 4.1,4.3-4.5), have been converted to a standard form of  $\mathbf{g}(\mathbf{x}) \leq 0$  and  $\mathbf{h}(\mathbf{x}) = 0$  (in Eqs. 4.8,4.10-4.16). The active set of inequality constraints is denoted as  $\mathbf{g}^+$ .

The penalty function employed in this paper uses a quadratic penalty with the penalty schedule linked to SBO iteration number

$$\Phi(\mathbf{x}, r_p) = f(\mathbf{x}) + r_p \mathbf{g}^+(\mathbf{x})^T \mathbf{g}^+(\mathbf{x}) + r_p \mathbf{h}(\mathbf{x})^T \mathbf{h}(\mathbf{x}) \quad (4.8)$$

$$r_p = e^{(k+\text{offset})/10} \quad (4.9)$$

The adaptive penalty function is identical in form to Eq. 4.8, but adapts  $r_p$  using monotonic increases in the iteration offset value in order to accept any iterate that reduces the constraint violation.

The Lagrangian merit function is

$$\Phi(\mathbf{x}, \lambda_g, \lambda_h) = f(\mathbf{x}) + \lambda_g^T \mathbf{g}^+(\mathbf{x}) + \lambda_h^T \mathbf{h}(\mathbf{x}) \quad (4.10)$$

for which the Lagrange multiplier estimation is discussed in Section 4.3. Away from the optimum, it is possible for the least squares estimates of the Lagrange multipliers for active constraints to be zero, which equates to omitting the contribution of an active constraint from the merit function. This is undesirable for tracking SBO progress, so usage of the Lagrangian merit function is normally restricted to approximate subproblems and hard convergence assessments.

The augmented Lagrangian employed in this paper follows the sign conventions described in [67]

$$\Phi(\mathbf{x}, \lambda_\psi, \lambda_h, r_p) = f(\mathbf{x}) + \lambda_\psi^T \psi(\mathbf{x}) + r_p \psi(\mathbf{x})^T \psi(\mathbf{x}) + \lambda_h^T \mathbf{h}(\mathbf{x}) + r_p \mathbf{h}(\mathbf{x})^T \mathbf{h}(\mathbf{x}) \quad (4.11)$$

$$\psi_i = \max \left\{ g_i, -\frac{\lambda_{\psi_i}}{2r_p} \right\} \quad (4.12)$$

where  $\psi(\mathbf{x})$  is derived from the elimination of slack variables for the inequality constraints. In this case, simple zeroth-order Lagrange multiplier updates may be used:

$$\lambda_\psi^{k+1} = \lambda_\psi^k + 2r_p \psi(\mathbf{x}) \quad (4.13)$$

$$\lambda_h^{k+1} = \lambda_h^k + 2r_p \mathbf{h}(\mathbf{x}) \quad (4.14)$$

The updating of multipliers and penalties is carefully orchestrated [13] to drive reduction in constraint violation of the iterates. The penalty updates can be more conservative than in Eq. 4.9, often using an infrequent application of a constant multiplier rather than a fixed exponential progression.

### 4.3 Convergence assessment

To terminate the SBO process, hard and soft convergence metrics are monitored. It is preferable for SBO studies to satisfy hard convergence metrics, but this is not always practical (e.g., when gradients are unavailable or unreliable). Therefore, simple soft convergence criteria are also employed which monitor for diminishing returns (relative improvement in the merit function less than a tolerance for some number of consecutive iterations).

To assess hard convergence, one calculates the norm of the projected gradient of a merit function whenever the feasibility tolerance is satisfied. The best merit function for this purpose is the Lagrangian merit function from Eq. 4.10. This requires a least squares estimation for the Lagrange multipliers that best minimize the projected gradient:

$$\nabla_x \Phi(\mathbf{x}, \boldsymbol{\lambda}_g, \boldsymbol{\lambda}_h) = \nabla_x f(\mathbf{x}) + \boldsymbol{\lambda}_g^T \nabla_x \mathbf{g}^+(\mathbf{x}) + \boldsymbol{\lambda}_h^T \nabla_x \mathbf{h}(\mathbf{x}) \quad (4.15)$$

where gradient portions directed into active global variable bounds have been removed. This can be posed as a linear least squares problem for the multipliers:

$$\mathbf{A}\boldsymbol{\lambda} = -\nabla_x f \quad (4.16)$$

where  $\mathbf{A}$  is the matrix of active constraint gradients,  $\boldsymbol{\lambda}_g$  is constrained to be non-negative, and  $\boldsymbol{\lambda}_h$  is unrestricted in sign. To estimate the multipliers using non-negative and bound-constrained linear least squares, the NNLS and BVLS routines [46] from NETLIB are used, respectively.

### 4.4 Constraint relaxation

The goal of constraint relaxation is to achieve efficiency through the balance of feasibility and optimality when the trust region restrictions prevent the location of feasible solutions to constrained approximate subproblems (Eqs. 4.3-4.5, and other formulations from rows 2-3 in Table 4.1). The SBO algorithm starting from infeasible points will commonly generate iterates which seek to satisfy feasibility conditions without regard to objective reduction [52].

One approach for achieving this balance is to use *relaxed constraints* when iterates are infeasible with respect to the surrogate constraints. We follow Perez, Renaud, and Watson [53], and use a *global homotopy* mapping the relaxed constraints and the surrogate constraints. For formulations in Eqs. 4.3 and 4.5 (and others from row 3 in Table 4.1), the relaxed constraints are defined from

$$\tilde{\mathbf{g}}^k(\mathbf{x}, \tau) = \hat{\mathbf{g}}^k(\mathbf{x}) + (1 - \tau)\mathbf{b}_g \quad (4.17)$$

$$\tilde{\mathbf{h}}^k(\mathbf{x}, \tau) = \hat{\mathbf{h}}^k(\mathbf{x}) + (1 - \tau)\mathbf{b}_h \quad (4.18)$$

For Eq. 4.4 (and others from row 2 in Table 4.1), the original surrogate constraints  $\hat{\mathbf{g}}^k(\mathbf{x})$  and  $\hat{\mathbf{h}}^k(\mathbf{x})$  in Eqs. 4.17-4.18 are replaced with their linearized forms  $(\hat{\mathbf{g}}^k(\mathbf{x}_c^k) + \nabla \hat{\mathbf{g}}^k(\mathbf{x}_c^k)^T(\mathbf{x} - \mathbf{x}_c^k))$  and  $(\hat{\mathbf{h}}^k(\mathbf{x}_c^k) + \nabla \hat{\mathbf{h}}^k(\mathbf{x}_c^k)^T(\mathbf{x} - \mathbf{x}_c^k))$ , respectively). The approximate subproblem is then reposed using the relaxed constraints as

$$\begin{aligned} & \text{minimize} && \hat{f}^k(\mathbf{x}) \text{ or } \hat{\Phi}^k(\mathbf{x}) \\ & \text{subject to} && \mathbf{g}_l \leq \tilde{\mathbf{g}}^k(\mathbf{x}, \tau^k) \leq \mathbf{g}_u \\ & && \tilde{\mathbf{h}}^k(\mathbf{x}, \tau^k) = \mathbf{h}_t \\ & && \|\mathbf{x} - \mathbf{x}_c^k\|_\infty \leq \Delta^k \end{aligned} \quad (4.19)$$

in place of the corresponding subproblems in Eqs. 4.3-4.5. Alternatively, since the relaxation terms are constants for the  $k^{th}$  iteration, it may be more convenient for the implementation to constrain  $\hat{\mathbf{g}}^k(\mathbf{x})$  and  $\hat{\mathbf{h}}^k(\mathbf{x})$  (or their

linearized forms) subject to relaxed bounds and targets  $(\tilde{\mathbf{g}}_l^k, \tilde{\mathbf{g}}_u^k, \tilde{\mathbf{h}}_t^k)$ . The parameter  $\tau$  is the homotopy parameter controlling the extent of the relaxation: when  $\tau = 0$ , the constraints are fully relaxed, and when  $\tau = 1$ , the surrogate constraints are recovered. The vectors  $\mathbf{b}_g, \mathbf{b}_h$  are chosen so that the starting point,  $\mathbf{x}^0$ , is feasible with respect to the fully relaxed constraints:

$$\mathbf{g}_l \leq \tilde{\mathbf{g}}^0(\mathbf{x}^0, 0) \leq \mathbf{g}_u \quad (4.20)$$

$$\tilde{\mathbf{h}}^0(\mathbf{x}^0, 0) = \mathbf{h}_t \quad (4.21)$$

At the start of the SBO algorithm,  $\tau^0 = 0$  if  $\mathbf{x}^0$  is infeasible with respect to the unrelaxed surrogate constraints; otherwise  $\tau^0 = 1$  (i.e., no constraint relaxation is used). At the start of the  $k^{th}$  SBO iteration where  $\tau^{k-1} < 1$ ,  $\tau^k$  is determined by solving the subproblem

$$\begin{aligned} & \text{maximize} && \tau^k \\ & \text{subject to} && \mathbf{g}_l \leq \tilde{\mathbf{g}}^k(\mathbf{x}, \tau^k) \leq \mathbf{g}_u \\ & && \tilde{\mathbf{h}}^k(\mathbf{x}, \tau^k) = \mathbf{h}_t \\ & && \|\mathbf{x} - \mathbf{x}_c^k\|_\infty \leq \Delta^k \\ & && \tau^k \geq 0 \end{aligned} \quad (4.22)$$

starting at  $(\mathbf{x}_*^{k-1}, \tau^{k-1})$ , and then adjusted as follows:

$$\tau^k = \min \{1, \tau^{k-1} + \alpha (\tau_{\max}^k - \tau^{k-1})\} \quad (4.23)$$

The adjustment parameter  $0 < \alpha < 1$  is chosen so that the feasible region with respect to the relaxed constraints has positive volume within the trust region. Determining the optimal value for  $\alpha$  remains an open question and will be explored in future work.

After  $\tau^k$  is determined using this procedure, the problem in Eq. 4.19 is solved for  $\mathbf{x}_*^k$ . If the step is accepted, then the value of  $\tau^k$  is updated using the current iterate  $\mathbf{x}_*^k$  and the validated constraints  $\mathbf{g}(\mathbf{x}_*^k)$  and  $\mathbf{h}(\mathbf{x}_*^k)$ :

$$\tau^k = \min \{1, \min_i \tau_i, \min_j \tau_j\} \quad (4.24)$$

$$\text{where } \tau_i = 1 + \frac{\min\{g_i(\mathbf{x}_*^k) - g_{l_i}, g_{u_i} - g_i(\mathbf{x}_*^k)\}}{b_{g_i}} \quad (4.25)$$

$$\tau_j = 1 - \frac{|h_j(\mathbf{x}_*^k) - h_{t_j}|}{b_{h_j}} \quad (4.26)$$

Figure 4.2 illustrates the SBO algorithm on a two-dimensional problem with one inequality constraint starting from an infeasible point,  $\mathbf{x}^0$ . The minimizer of the problem is denoted as  $\mathbf{x}^*$ . Iterates generated using the surrogate constraints are shown in red, where feasibility is achieved first, and then progress is made toward the optimal point. The iterates generated using the relaxed constraints are shown in blue, where a balance of satisfying feasibility and optimality has been achieved, leading to fewer overall SBO iterations.

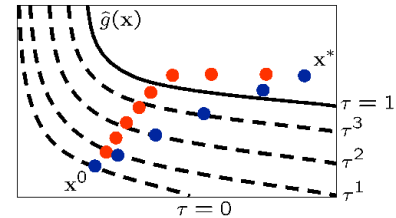


Figure 4.2: Illustration of SBO iterates using surrogate (red) and relaxed (blue) constraints.

The behavior illustrated in Fig. 4.2 is an example where using the relaxed constraints over the surrogate constraints may improve the overall performance of the SBO algorithm by reducing the number of iterations performed. This improvement comes at the cost of solving the minimization subproblem in Eq. 4.22, which can be significant in some cases (i.e., when the cost of evaluating  $\tilde{\mathbf{g}}^k(\mathbf{x})$  and  $\tilde{\mathbf{h}}^k(\mathbf{x})$  is not negligible, such as with multifidelity or ROM surrogates). As shown in the numerical experiments involving the Barnes

problem presented in [53], the directions toward constraint violation reduction and objective function reduction may be in opposing directions. In such cases, the use of the relaxed constraints may result in an *increase* in the overall number of SBO iterations since feasibility must ultimately take precedence.





## Chapter 5

# Optimization Under Uncertainty (OUU)

### 5.1 Reliability-Based Design Optimization (RBDO)

Reliability-based design optimization (RBDO) methods are used to perform design optimization accounting for reliability metrics. The reliability analysis capabilities described in Section 1.1 provide a substantial foundation for exploring a variety of gradient-based RBDO formulations. [20] investigated bi-level, fully-analytic bi-level, and first-order sequential RBDO approaches employing underlying first-order reliability assessments. [21] investigated fully-analytic bi-level and second-order sequential RBDO approaches employing underlying second-order reliability assessments. These methods are overviewed in the following sections.

#### 5.1.1 Bi-level RBDO

The simplest and most direct RBDO approach is the bi-level approach in which a full reliability analysis is performed for every optimization function evaluation. This involves a nesting of two distinct levels of optimization within each other, one at the design level and one at the MPP search level.

Since an RBDO problem will typically specify both the  $\bar{z}$  level and the  $\bar{p}/\bar{\beta}$  level, one can use either the RIA or the PMA formulation for the UQ portion and then constrain the result in the design optimization portion. In particular, RIA reliability analysis maps  $\bar{z}$  to  $p/\beta$ , so RIA RBDO constrains  $p/\beta$ :

$$\begin{aligned} & \text{minimize} && f \\ & \text{subject to} && \beta \geq \bar{\beta} \\ & && \text{or } p \leq \bar{p} \end{aligned} \tag{5.1}$$

And PMA reliability analysis maps  $\bar{p}/\bar{\beta}$  to  $z$ , so PMA RBDO constrains  $z$ :

$$\begin{aligned} & \text{minimize} && f \\ & \text{subject to} && z \geq \bar{z} \end{aligned} \tag{5.2}$$

where  $z \geq \bar{z}$  is used as the RBDO constraint for a cumulative failure probability (failure defined as  $z \leq \bar{z}$ ) but  $z \leq \bar{z}$  would be used as the RBDO constraint for a complementary cumulative failure probability (failure defined as  $z \geq \bar{z}$ ). It is worth noting that DAKOTA is not limited to these types of inequality-constrained RBDO formulations; rather, they are convenient examples. DAKOTA supports general optimization under uncertainty

mappings [25] which allow flexible use of statistics within multiple objectives, inequality constraints, and equality constraints.

An important performance enhancement for bi-level methods is the use of sensitivity analysis to analytically compute the design gradients of probability, reliability, and response levels. When design variables are separate from the uncertain variables (i.e., they are not distribution parameters), then the following first-order expressions may be used [39, 45, 4]:

$$\nabla_{\mathbf{d}} z = \nabla_{\mathbf{d}} g \quad (5.3)$$

$$\nabla_{\mathbf{d}} \beta_{cdf} = \frac{1}{\|\nabla_{\mathbf{u}} G\|} \nabla_{\mathbf{d}} g \quad (5.4)$$

$$\nabla_{\mathbf{d}} p_{cdf} = -\phi(-\beta_{cdf}) \nabla_{\mathbf{d}} \beta_{cdf} \quad (5.5)$$

where it is evident from Eqs. 1.10-1.11 that  $\nabla_{\mathbf{d}} \beta_{cdf} = -\nabla_{\mathbf{d}} \beta_{cdf}$  and  $\nabla_{\mathbf{d}} p_{cdf} = -\nabla_{\mathbf{d}} p_{cdf}$ . In the case of second-order integrations, Eq. 5.5 must be expanded to include the curvature correction. For Breitung's correction (Eq. 1.37),

$$\nabla_{\mathbf{d}} p_{cdf} = \left[ \Phi(-\beta_p) \sum_{i=1}^{n-1} \left( \frac{-\kappa_i}{2(1 + \beta_p \kappa_i)^{\frac{3}{2}}} \prod_{\substack{j=1 \\ j \neq i}}^{n-1} \frac{1}{\sqrt{1 + \beta_p \kappa_j}} \right) - \phi(-\beta_p) \prod_{i=1}^{n-1} \frac{1}{\sqrt{1 + \beta_p \kappa_i}} \right] \nabla_{\mathbf{d}} \beta_{cdf} \quad (5.6)$$

where  $\nabla_{\mathbf{d}} \kappa_i$  has been neglected and  $\beta_p \geq 0$  (see Section 1.1.2.2). Other approaches assume the curvature correction is nearly independent of the design variables [54], which is equivalent to neglecting the first term in Eq. 5.6.

To capture second-order probability estimates within an RIA RBDO formulation using well-behaved  $\beta$  constraints, a generalized reliability index can be introduced where, similar to Eq. 1.8,

$$\beta_{cdf}^* = -\Phi^{-1}(p_{cdf}) \quad (5.7)$$

for second-order  $p_{cdf}$ . This reliability index is no longer equivalent to the magnitude of  $\mathbf{u}$ , but rather is a convenience metric for capturing the effect of more accurate probability estimates. The corresponding generalized reliability index sensitivity, similar to Eq. 5.5, is

$$\nabla_{\mathbf{d}} \beta_{cdf}^* = -\frac{1}{\phi(-\beta_{cdf}^*)} \nabla_{\mathbf{d}} p_{cdf} \quad (5.8)$$

where  $\nabla_{\mathbf{d}} p_{cdf}$  is defined from Eq. 5.6. Even when  $\nabla_{\mathbf{d}} g$  is estimated numerically, Eqs. 5.3-5.8 can be used to avoid numerical differencing across full reliability analyses.

When the design variables are distribution parameters of the uncertain variables,  $\nabla_{\mathbf{d}} g$  is expanded with the chain rule and Eqs. 5.3 and 5.4 become

$$\nabla_{\mathbf{d}} z = \nabla_{\mathbf{d}} \mathbf{x} \nabla_{\mathbf{x}} g \quad (5.9)$$

$$\nabla_{\mathbf{d}} \beta_{cdf} = \frac{1}{\|\nabla_{\mathbf{u}} G\|} \nabla_{\mathbf{d}} \mathbf{x} \nabla_{\mathbf{x}} g \quad (5.10)$$

where the design Jacobian of the transformation ( $\nabla_{\mathbf{d}} \mathbf{x}$ ) may be obtained analytically for uncorrelated  $\mathbf{x}$  or semi-analytically for correlated  $\mathbf{x}$  ( $\nabla_{\mathbf{d}} \mathbf{L}$  is evaluated numerically) by differentiating Eqs. 1.13 and 1.14 with respect to the distribution parameters. Eqs. 5.5-5.8 remain the same as before. For this design variable case, all required information for the sensitivities is available from the MPP search.

Since Eqs. 5.3-5.10 are derived using the Karush-Kuhn-Tucker optimality conditions for a converged MPP, they are appropriate for RBDO using AMV+, AMV<sup>2</sup>+, TANA, FORM, and SORM, but not for RBDO using MV-FOSM, MVSOSM, AMV, or AMV<sup>2</sup>.

### 5.1.2 Sequential/Surrogate-based RBDO

An alternative RBDO approach is the sequential approach, in which additional efficiency is sought through breaking the nested relationship of the MPP and design searches. The general concept is to iterate between optimization and uncertainty quantification, updating the optimization goals based on the most recent probabilistic assessment results. This update may be based on safety factors [72] or other approximations [19].

A particularly effective approach for updating the optimization goals is to use the  $p/\beta/z$  sensitivity analysis of Eqs. 5.3-5.10 in combination with local surrogate models [79]. In [20] and [21], first-order and second-order Taylor series approximations were employed within a trust-region model management framework [36] in order to adaptively manage the extent of the approximations and ensure convergence of the RBDO process. Surrogate models were used for both the objective function and the constraints, although the use of constraint surrogates alone is sufficient to remove the nesting.

In particular, RIA trust-region surrogate-based RBDO employs surrogate models of  $f$  and  $p/\beta$  within a trust region  $\Delta^k$  centered at  $\mathbf{d}_c$ . For first-order surrogates:

$$\begin{aligned} & \text{minimize} && f(\mathbf{d}_c) + \nabla_{\mathbf{d}} f(\mathbf{d}_c)^T (\mathbf{d} - \mathbf{d}_c) \\ & \text{subject to} && \beta(\mathbf{d}_c) + \nabla_{\mathbf{d}} \beta(\mathbf{d}_c)^T (\mathbf{d} - \mathbf{d}_c) \geq \bar{\beta} \\ & && \text{or } p(\mathbf{d}_c) + \nabla_{\mathbf{d}} p(\mathbf{d}_c)^T (\mathbf{d} - \mathbf{d}_c) \leq \bar{p} \\ & && \|\mathbf{d} - \mathbf{d}_c\|_{\infty} \leq \Delta^k \end{aligned} \quad (5.11)$$

and for second-order surrogates:

$$\begin{aligned} & \text{minimize} && f(\mathbf{d}_c) + \nabla_{\mathbf{d}} f(\mathbf{d}_c)^T (\mathbf{d} - \mathbf{d}_c) + \frac{1}{2} (\mathbf{d} - \mathbf{d}_c)^T \nabla_{\mathbf{d}}^2 f(\mathbf{d}_c) (\mathbf{d} - \mathbf{d}_c) \\ & \text{subject to} && \beta(\mathbf{d}_c) + \nabla_{\mathbf{d}} \beta(\mathbf{d}_c)^T (\mathbf{d} - \mathbf{d}_c) + \frac{1}{2} (\mathbf{d} - \mathbf{d}_c)^T \nabla_{\mathbf{d}}^2 \beta(\mathbf{d}_c) (\mathbf{d} - \mathbf{d}_c) \geq \bar{\beta} \\ & && \text{or } p(\mathbf{d}_c) + \nabla_{\mathbf{d}} p(\mathbf{d}_c)^T (\mathbf{d} - \mathbf{d}_c) + \frac{1}{2} (\mathbf{d} - \mathbf{d}_c)^T \nabla_{\mathbf{d}}^2 p(\mathbf{d}_c) (\mathbf{d} - \mathbf{d}_c) \leq \bar{p} \\ & && \|\mathbf{d} - \mathbf{d}_c\|_{\infty} \leq \Delta^k \end{aligned} \quad (5.12)$$

For PMA trust-region surrogate-based RBDO, surrogate models of  $f$  and  $z$  are employed within a trust region  $\Delta^k$  centered at  $\mathbf{d}_c$ . For first-order surrogates:

$$\begin{aligned} & \text{minimize} && f(\mathbf{d}_c) + \nabla_{\mathbf{d}} f(\mathbf{d}_c)^T (\mathbf{d} - \mathbf{d}_c) \\ & \text{subject to} && z(\mathbf{d}_c) + \nabla_{\mathbf{d}} z(\mathbf{d}_c)^T (\mathbf{d} - \mathbf{d}_c) \geq \bar{z} \\ & && \|\mathbf{d} - \mathbf{d}_c\|_{\infty} \leq \Delta^k \end{aligned} \quad (5.13)$$

and for second-order surrogates:

$$\begin{aligned} & \text{minimize} && f(\mathbf{d}_c) + \nabla_{\mathbf{d}} f(\mathbf{d}_c)^T (\mathbf{d} - \mathbf{d}_c) + \frac{1}{2} (\mathbf{d} - \mathbf{d}_c)^T \nabla_{\mathbf{d}}^2 f(\mathbf{d}_c) (\mathbf{d} - \mathbf{d}_c) \\ & \text{subject to} && z(\mathbf{d}_c) + \nabla_{\mathbf{d}} z(\mathbf{d}_c)^T (\mathbf{d} - \mathbf{d}_c) + \frac{1}{2} (\mathbf{d} - \mathbf{d}_c)^T \nabla_{\mathbf{d}}^2 z(\mathbf{d}_c) (\mathbf{d} - \mathbf{d}_c) \geq \bar{z} \\ & && \|\mathbf{d} - \mathbf{d}_c\|_{\infty} \leq \Delta^k \end{aligned} \quad (5.14)$$

where the sense of the  $z$  constraint may vary as described previously. The second-order information in Eqs. 5.12 and 5.14 will typically be approximated with quasi-Newton updates.

## 5.2 Stochastic Expansion-Based Design Optimization (SEBDO)

### 5.2.1 Stochastic Sensitivity Analysis

Section 2.10 describes sensitivity analysis of the polynomial chaos expansion with respect to the expansion variables. Here we extend this analysis to include sensitivity analysis of probabilistic moments with respect to non-

probabilistic (i.e., design or epistemic uncertain) variables.

### 5.2.1.1 Local sensitivity analysis: first-order probabilistic expansions

With the introduction of nonprobabilistic variables  $s$  (for example, design variables or epistemic uncertain variables), a polynomial chaos expansion only over the probabilistic variables  $\xi$  has the functional relationship:

$$R(\xi, s) \cong \sum_{j=0}^P \alpha_j(s) \Psi_j(\xi) \quad (5.15)$$

For computing sensitivities of response mean and variance, the  $ij$  indices may be dropped from Eqs. 2.28 and 2.29, simplifying to

$$\mu(s) = \alpha_0(s), \quad \sigma^2(s) = \sum_{k=1}^P \alpha_k^2(s) \langle \Psi_k^2 \rangle \quad (5.16)$$

Sensitivities of Eq. 5.16 with respect to the nonprobabilistic variables are as follows, where independence of  $s$  and  $\xi$  is assumed:

$$\frac{d\mu}{ds} = \frac{d\alpha_0}{ds} = \left\langle \frac{dR}{ds} \right\rangle \quad (5.17)$$

$$\frac{d\sigma^2}{ds} = \sum_{k=1}^P \langle \Psi_k^2 \rangle \frac{d\alpha_k^2}{ds} = 2 \sum_{k=1}^P \alpha_k \left\langle \frac{dR}{ds}, \Psi_k \right\rangle \quad (5.18)$$

where

$$\frac{d\alpha_k}{ds} = \frac{\left\langle \frac{dR}{ds}, \Psi_k \right\rangle}{\langle \Psi_k^2 \rangle} \quad (5.19)$$

has been used. Due to independence, the coefficients calculated in Eq. 5.19 may be interpreted as either the derivatives of the expectations or the expectations of the derivatives, or more precisely, the nonprobabilistic sensitivities of the chaos coefficients for the response expansion or the chaos coefficients of an expansion for the nonprobabilistic sensitivities of the response. The evaluation of integrals involving  $\frac{dR}{ds}$  extends the data requirements for the PCE approach to include response sensitivities at each of the sampled points. The resulting expansions are valid only for a particular set of nonprobabilistic variables and must be recalculated each time the nonprobabilistic variables are modified.

Similarly for stochastic collocation,

$$R(\xi, s) \cong \sum_{k=1}^{N_p} r_k(s) L_k(\xi) \quad (5.20)$$

leads to

$$\mu(s) = \sum_{k=1}^{N_p} r_k(s) w_k, \quad \sigma^2(s) = \sum_{k=1}^{N_p} r_k^2(s) w_k - \mu^2(s) \quad (5.21)$$

$$\frac{d\mu}{ds} = \sum_{k=1}^{N_p} w_k \frac{dr_k}{ds} \quad (5.22)$$

$$\frac{d\sigma^2}{ds} = \sum_{k=1}^{N_p} 2w_k r_k \frac{dr_k}{ds} - 2\mu \frac{d\mu}{ds} = \sum_{k=1}^{N_p} 2w_k (r_k - \mu) \frac{dr_k}{ds} \quad (5.23)$$

### 5.2.1.2 Local sensitivity analysis: zeroth-order combined expansions

Alternatively, a stochastic expansion can be formed over both  $\xi$  and  $s$ . Assuming a bounded design domain  $s_L \leq s \leq s_U$  (with no implied probability content), a Legendre chaos basis would be appropriate for each of the dimensions in  $s$  within a polynomial chaos expansion.

$$R(\xi, s) \cong \sum_{j=0}^P \alpha_j \Psi_j(\xi, s) \quad (5.24)$$

In this case, design sensitivities for the mean and variance do not require response sensitivity data, but this comes at the cost of forming the PCE over additional dimensions. For this combined variable expansion, the mean and variance are evaluated by performing the expectations over only the probabilistic expansion variables, which eliminates the polynomial dependence on  $\xi$ , leaving behind the desired polynomial dependence of the moments on  $s$ :

$$\mu_R(s) = \sum_{j=0}^P \alpha_j \langle \Psi_j(\xi, s) \rangle_{\xi} \quad (5.25)$$

$$\sigma_R^2(s) = \sum_{j=0}^P \sum_{k=0}^P \alpha_j \alpha_k \langle \Psi_j(\xi, s) \Psi_k(\xi, s) \rangle_{\xi} - \mu_R^2(s) \quad (5.26)$$

The remaining polynomials may then be differentiated with respect to  $s$ . In this approach, the combined PCE is valid for the full design variable range ( $s_L \leq s \leq s_U$ ) and does not need to be updated for each change in nonprobabilistic variables, although adaptive localization techniques (i.e., trust region model management approaches) can be employed when improved local accuracy of the sensitivities is required.

Similarly for stochastic collocation,

$$R(\xi, s) \cong \sum_{j=1}^{N_p} r_j L_j(\xi, s) \quad (5.27)$$

leads to

$$\mu_R(s) = \sum_{j=1}^{N_p} r_j \langle L_j(\xi, s) \rangle_{\xi} \quad (5.28)$$

$$\sigma_R^2(s) = \sum_{j=1}^{N_p} \sum_{k=1}^{N_p} r_j r_k \langle L_j(\xi, s) L_k(\xi, s) \rangle_{\xi} - \mu_R^2(s) \quad (5.29)$$

where the remaining polynomials not eliminated by the expectation over  $\xi$  are again differentiated with respect to  $s$ .

### 5.2.1.3 Inputs and outputs

There are two types of nonprobabilistic variables for which sensitivities must be calculated: “augmented,” where the nonprobabilistic variables are separate from and augment the probabilistic variables, and “inserted,” where the nonprobabilistic variables define distribution parameters for the probabilistic variables. Any inserted nonprobabilistic variable sensitivities must be handled using Eqs. 5.17-5.18 and Eqs. 5.22-5.23 where  $\frac{dR}{ds}$  is calculated as

$\frac{dR}{dx} \frac{dx}{ds}$  and  $\frac{dx}{ds}$  is the Jacobian of the variable transformation  $\mathbf{x} = T^{-1}(\boldsymbol{\xi})$  with respect to the inserted nonprobabilistic variables. In addition, parameterized polynomials (generalized Gauss-Laguerre, Jacobi, and numerically-generated polynomials) may introduce a  $\frac{d\Psi}{ds}$  or  $\frac{dL}{ds}$  dependence for inserted  $s$  that will introduce additional terms in the sensitivity expressions.

While moment sensitivities directly enable robust design optimization and interval estimation formulations which seek to control or bound response variance, control or bounding of reliability requires sensitivities of tail statistics. In this work, the sensitivity of simple moment-based approximations to cumulative distribution function (CDF) and complementary cumulative distribution function (CCDF) mappings (Eqs. 1.3–1.4) are employed for this purpose, such that it is straightforward to form approximate design sensitivities of reliability index  $\beta$  (forward reliability mapping  $\bar{z} \rightarrow \beta$ ) or response level  $z$  (inverse reliability mapping  $\bar{\beta} \rightarrow z$ ) from the moment design sensitivities and the specified levels  $\bar{\beta}$  or  $\bar{z}$ .

## 5.2.2 Optimization Formulations

Given the capability to compute analytic statistics of the response along with design sensitivities of these statistics, DAKOTA supports bi-level, sequential, and multifidelity approaches for optimization under uncertainty (OUU). The latter two approaches apply surrogate modeling approaches (data fits and multifidelity modeling) to the uncertainty analysis and then apply trust region model management to the optimization process.

### 5.2.2.1 Bi-level SEBDO

The simplest and most direct approach is to employ these analytic statistics and their design derivatives from Section 5.2.1 directly within an optimization loop. This approach is known as bi-level OUU, since there is an inner level uncertainty analysis nested within an outer level optimization.

Consider the common reliability-based design example of a deterministic objective function with a reliability constraint:

$$\begin{aligned} & \text{minimize} && f \\ & \text{subject to} && \beta \geq \bar{\beta} \end{aligned} \tag{5.30}$$

where  $\beta$  is computed relative to a prescribed threshold response value  $\bar{z}$  (e.g., a failure threshold) and is constrained by a prescribed reliability level  $\bar{\beta}$  (minimum allowable reliability in the design), and is either a CDF or CCDF index depending on the definition of the failure domain (i.e., defined from whether the associated failure probability is cumulative,  $p(g \leq \bar{z})$ , or complementary cumulative,  $p(g > \bar{z})$ ).

Another common example is robust design in which the constraint enforcing a reliability lower-bound has been replaced with a constraint enforcing a variance upper-bound  $\bar{\sigma}^2$  (maximum allowable variance in the design):

$$\begin{aligned} & \text{minimize} && f \\ & \text{subject to} && \sigma^2 \leq \bar{\sigma}^2 \end{aligned} \tag{5.31}$$

Solving these problems using a bi-level approach involves computing  $\beta$  and  $\frac{d\beta}{ds}$  for Eq. 5.30 or  $\sigma^2$  and  $\frac{d\sigma^2}{ds}$  for Eq. 5.31 for each set of design variables  $s$  passed from the optimizer. This approach is supported for both probabilistic and combined expansions using PCE and SC.

### 5.2.2.2 Sequential/Surrogate-Based SEBDO

An alternative OUU approach is the sequential approach, in which additional efficiency is sought through breaking the nested relationship of the UQ and optimization loops. The general concept is to iterate between optimization and uncertainty quantification, updating the optimization goals based on the most recent uncertainty assessment results. This approach is common with the reliability methods community, for which the updating strategy may be based on safety factors [72] or other approximations [19].

A particularly effective approach for updating the optimization goals is to use data fit surrogate models, and in particular, local Taylor series models allow direct insertion of stochastic sensitivity analysis capabilities. In Ref. [20], first-order Taylor series approximations were explored, and in Ref. [21], second-order Taylor series approximations are investigated. In both cases, a trust-region model management framework [23] is used to adaptively manage the extent of the approximations and ensure convergence of the OUU process. Surrogate models are used for both the objective and the constraint functions, although the use of surrogates is only required for the functions containing statistical results; deterministic functions may remain explicit is desired.

In particular, trust-region surrogate-based optimization for reliability-based design employs surrogate models of  $f$  and  $\beta$  within a trust region  $\Delta^k$  centered at  $\mathbf{s}_c$ :

$$\begin{aligned} & \text{minimize} && f(\mathbf{s}_c) + \nabla_s f(\mathbf{s}_c)^T (\mathbf{s} - \mathbf{s}_c) \\ & \text{subject to} && \beta(\mathbf{s}_c) + \nabla_s \beta(\mathbf{s}_c)^T (\mathbf{s} - \mathbf{s}_c) \geq \bar{\beta} \\ & && \|\mathbf{s} - \mathbf{s}_c\|_\infty \leq \Delta^k \end{aligned} \quad (5.32)$$

and trust-region surrogate-based optimization for robust design employs surrogate models of  $f$  and  $\sigma^2$  within a trust region  $\Delta^k$  centered at  $\mathbf{s}_c$ :

$$\begin{aligned} & \text{minimize} && f(\mathbf{s}_c) + \nabla_s f(\mathbf{s}_c)^T (\mathbf{s} - \mathbf{s}_c) \\ & \text{subject to} && \sigma^2(\mathbf{s}_c) + \nabla_s \sigma^2(\mathbf{s}_c)^T (\mathbf{s} - \mathbf{s}_c) \leq \bar{\sigma}^2 \\ & && \|\mathbf{s} - \mathbf{s}_c\|_\infty \leq \Delta^k \end{aligned} \quad (5.33)$$

Second-order local surrogates may also be employed, where the Hessians are typically approximated from an accumulation of curvature information using quasi-Newton updates [50] such as Broyden-Fletcher-Goldfarb-Shanno (BFGS, Eq. 1.41) or symmetric rank one (SR1, Eq. 1.42). The sequential approach is available for probabilistic expansions using PCE and SC.

### 5.2.2.3 Multifidelity SEBDO

The multifidelity OUU approach is another trust-region surrogate-based approach. Instead of the surrogate UQ model being a simple data fit (in particular, first-/second-order Taylor series model) of the truth UQ model results, distinct UQ models of differing fidelity are now employed. This differing UQ fidelity could stem from the fidelity of the underlying simulation model, the fidelity of the UQ algorithm, or both. In this section, we focus on the fidelity of the UQ algorithm. For reliability methods, this could entail varying fidelity in approximating assumptions (e.g., Mean Value for low fidelity, SORM for high fidelity), and for stochastic expansion methods, it could involve differences in selected levels of  $p$  and  $h$  refinement.

Here, we define UQ fidelity as point-wise accuracy in the design space and take the high fidelity truth model to be the probabilistic expansion PCE/SC model, with validity only at a single design point. The low fidelity model, whose validity over the design space will be adaptively controlled, will be either the combined expansion PCE/SC model, with validity over a range of design parameters, or the MVFOSM reliability method, with validity only at a single design point. The combined expansion low fidelity approach will span the current trust region

of the design space and will be reconstructed for each new trust region. Trust region adaptation will ensure that the combined expansion approach remains sufficiently accurate for design purposes. By taking advantage of the design space spanning, one can eliminate the cost of multiple low fidelity UQ analyses within the trust region, with fallback to the greater accuracy and higher expense of the probabilistic expansion approach when needed. The MVFOSM low fidelity approximation must be reformed for each change in design variables, but it only requires a single evaluation of a response function and its derivative to approximate the response mean and variance from the input mean and covariance (Eqs. 1.1–1.2) from which forward/inverse CDF/CCDF reliability mappings can be generated using Eqs. 1.3–1.4. This is the least expensive UQ option, but its limited accuracy<sup>1</sup> may dictate the use of small trust regions, resulting in greater iterations to convergence. The expense of optimizing a combined expansion, on the other hand, is not significantly less than that of optimizing the high fidelity UQ model, but its representation of global trends should allow the use of larger trust regions, resulting in reduced iterations to convergence. The design derivatives of each of the PCE/SC expansion models provide the necessary data to correct the low fidelity model to first-order consistency with the high fidelity model at the center of each trust region, ensuring convergence of the multifidelity optimization process to the high fidelity optimum. Design derivatives of the MVFOSM statistics are currently evaluated numerically using forward finite differences.

Multifidelity optimization for reliability-based design can be formulated as:

$$\begin{aligned} & \text{minimize} && f(\mathbf{s}) \\ & \text{subject to} && \hat{\beta}_{hi}(\mathbf{s}) \geq \bar{\beta} \\ & && \|\mathbf{s} - \mathbf{s}_c\|_{\infty} \leq \Delta^k \end{aligned} \quad (5.34)$$

and multifidelity optimization for robust design can be formulated as:

$$\begin{aligned} & \text{minimize} && f(\mathbf{s}) \\ & \text{subject to} && \hat{\sigma}_{hi}^2(\mathbf{s}) \leq \bar{\sigma}^2 \\ & && \|\mathbf{s} - \mathbf{s}_c\|_{\infty} \leq \Delta^k \end{aligned} \quad (5.35)$$

where the deterministic objective function is not approximated and  $\hat{\beta}_{hi}$  and  $\hat{\sigma}_{hi}^2$  are the approximated high-fidelity UQ results resulting from correction of the low-fidelity UQ results. In the case of an additive correction function:

$$\hat{\beta}_{hi}(\mathbf{s}) = \beta_{lo}(\mathbf{s}) + \alpha_{\beta}(\mathbf{s}) \quad (5.36)$$

$$\hat{\sigma}_{hi}^2(\mathbf{s}) = \sigma_{lo}^2(\mathbf{s}) + \alpha_{\sigma^2}(\mathbf{s}) \quad (5.37)$$

where correction functions  $\alpha(\mathbf{s})$  enforcing first-order consistency [24] are typically employed. Quasi-second-order correction functions [24] can also be employed, but care must be taken due to the different rates of curvature accumulation between the low and high fidelity models. In particular, since the low fidelity model is evaluated more frequently than the high fidelity model, it accumulates curvature information more quickly, such that enforcing quasi-second-order consistency with the high fidelity model can be detrimental in the initial iterations of the algorithm<sup>2</sup>. Instead, this consistency should only be enforced when sufficient high fidelity curvature information has been accumulated (e.g., after  $n$  rank one updates).

<sup>1</sup>MVFOSM is exact for linear functions with Gaussian inputs, but quickly degrades for nonlinear and/or non-Gaussian.

<sup>2</sup>Analytic and numerical Hessians, when available, are instantaneous with no accumulation rate concerns.



# Bibliography

- [1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, New York, 1965. [19](#), [24](#), [29](#)
- [2] B. M. Adams, W. J. Bohnhoff, K. R. Dalbey, J. P. Eddy, M. S. Eldred, D. M. Gay, K. Haskell, P. D. Hough, and L. P. Swiler. DAKOTA, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 5.0 reference manual. Technical Report SAND2010-2184, Sandia National Laboratories, Albuquerque, NM, 2010. Available online from <http://www.cs.sandia.gov/dakota/documentation.html>. [37](#)
- [3] N. M. Alexandrov, R. M. Lewis, C. R. Gumbert, L. L. Green, and P. A. Newman. Optimization with variable-fidelity models applied to wing design. In *Proceedings of the 38th Aerospace Sciences Meeting and Exhibit*, Reno, NV, 2000. AIAA Paper 2000-0841. [35](#), [36](#), [37](#)
- [4] M. Allen and K. Maute. Reliability-based design optimization of aeroelastic structures. *Struct. Multidiscip. O.*, 27:228–242, 2004. [44](#)
- [5] R. Askey and J. Wilson. Some basic hypergeometric polynomials that generalize jacobi polynomials. In *Mem. Amer. Math. Soc.* 319, Providence, RI, 1985. AMS. [19](#)
- [6] V. Barthelmann, E. Novak, and K. Ritter. High dimensional polynomial interpolation on sparse grids. *Adv. Comput. Math.*, 12(4):273–288, 2000. Multivariate polynomial interpolation. [26](#)
- [7] B. J. Bichon, M. S. Eldred, L. P. Swiler, S. Mahadevan, and J. M. McFarland. Efficient global reliability analysis for nonlinear implicit performance functions. *AIAA Journal*, 46(10):2459–2468, 2008. [13](#)
- [8] G. E. P. Box and D. R. Cox. An analysis of transformations. *J. Royal Stat. Soc., Series B*, 26:211–252, 1964. [8](#), [23](#)
- [9] K. Breitung. Asymptotic approximation for multinormal integrals. *J. Eng. Mech., ASCE*, 110(3):357–366, 1984. [11](#)
- [10] J. Burkardt. The “combining coefficient” for anisotropic sparse grids. Technical report, Virginia Tech, Blacksburg, VA, 2009. <http://people.sc.fsu.edu/~burkardt/presentations/sgmga-coefficient.pdf>. [27](#)
- [11] M. R. Celis, J. .E. Dennis, and R. .A. Tapia. A trust region strategy for nonlinear equality constrained optimization. In P. .T. Boggs, R. H. Byrd, and R. B. Schnabel, editors, *Numerical Optimization 1984*, pages 71–82. SIAM, Philadelphia, USA, 1985. [37](#)
- [12] X. Chen and N.C. Lind. Fast probability integration by three-parameter normal tail approximation. *Struct. Saf.*, 1:269–276, 1983. [8](#)

- [13] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust-Region Methods*. MPS-SIAM Series on Optimization, SIAM-MPS, Philadelphia, 2000. [38](#)
- [14] P. G. Constantine and M. S. Eldred. Sparse polynomial chaos expansions. *International Journal for Uncertainty Quantification*, in preparation. [27](#)
- [15] P. G. Constantine, D. F. Gleich, and G. Iaccarino. Spectral methods for parameterized matrix equations. *SIAM Journal on Matrix Analysis and Applications*, 31(5):2681–2699. [26](#)
- [16] N. Cressie. *Statistics of Spatial Data*. John Wiley and Sons, New York, 1991. [15](#)
- [17] A. Der Kiureghian and P. L. Liu. Structural reliability under incomplete information. *J. Eng. Mech., ASCE*, 112(EM-1):85–104, 1986. [8](#), [9](#), [20](#), [23](#), [24](#)
- [18] A. Dey and S. Mahadevan. Ductile structural system reliability analysis using adaptive importance sampling. *Structural Safety*, 20:137–154, 1998. [14](#)
- [19] X. Du and W. Chen. Sequential optimization and reliability assessment method for efficient probabilistic design. *J. Mech. Design*, 126:225–233, 2004. [45](#), [49](#)
- [20] M. S. Eldred, H. Agarwal, V. M. Perez, Jr. Wojtkiewicz, S. F., and J. E. Renaud. Investigation of reliability method formulations in dakota/uq. *Structure & Infrastructure Engineering: Maintenance, Management, Life-Cycle Design & Performance*, 3(3):199–213, 2007. [9](#), [13](#), [43](#), [45](#), [49](#)
- [21] M. S. Eldred and B. J. Bichon. Second-order reliability formulations in DAKOTA/UQ. In *Proceedings of the 47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, number AIAA-2006-1828, Newport, RI, May 1–4 2006. [9](#), [12](#), [43](#), [45](#), [49](#)
- [22] M. S. Eldred and J. Burkardt. Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification. In *Proceedings of the 47th AIAA Aerospace Sciences Meeting and Exhibit*, number AIAA-2009-0976, Orlando, FL, January 5–8, 2009. [25](#), [26](#), [27](#)
- [23] M. S. Eldred and D. M. Dunlavy. Formulations for surrogate-based optimization with data fit, multifidelity, and reduced-order models. In *Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, number AIAA-2006-7117, Portsmouth, VA, September 6–8 2006. [49](#)
- [24] M. S. Eldred, A. A. Giunta, and S. S. Collis. Second-order corrections for surrogate-based optimization with model hierarchies. In *Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, NY, Aug. 30–Sept. 1, 2004. AIAA Paper 2004-4457. [50](#)
- [25] M. S. Eldred, A. A. Giunta, S. F. Wojtkiewicz Jr., and T. G. Trucano. Formulations for surrogate-based optimization under uncertainty. In *Proc. 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, number AIAA-2002-5585, Atlanta, GA, September 4–6, 2002. [44](#)
- [26] M. S. Eldred, C. G. Webster, and P. Constantine. Evaluation of non-intrusive approaches for wiener-askey generalized polynomial chaos. In *Proceedings of the 10th AIAA Non-Deterministic Approaches Conference*, number AIAA-2008-1892, Schaumburg, IL, April 7–10 2008. [20](#)
- [27] G. M. Fadel, M. F. Riley, and J.-F. M. Barthelemy. Two point exponential approximation method for structural optimization. *Structural Optimization*, 2(2):117–124, 1990. [10](#)
- [28] R. Fletcher, S. Leyffer, and P. L. Toint. On the global convergence of a filter-SQP algorithm. *SIAM J. Optim.*, 13(1):44–59, 2002. [37](#), [38](#)
- [29] P. Frauenfelder, C. Schwab, and R. A. Todor. Finite elements for elliptic problems with stochastic coefficients. *Comput. Methods Appl. Mech. Engrg.*, 194(2-5):205–228, 2005. [26](#)

- [30] J. Gablonsky. Direct version 2.0 userguide technical report. Technical Report CRSC-TR01-08, North Carolina State University, Center for Research in Scientific Computation, Raleigh, NC, 2001. [16](#)
- [31] W. Gautschi. *Orthogonal Polynomials: Computation and Approximation*. Oxford University Press, New York, 2004. [20](#)
- [32] T. Gerstner and M. Griebel. Numerical integration using sparse grids. *Numer. Algorithms*, 18(3-4):209–232, 1998. [26](#)
- [33] T. Gerstner and M. Griebel. Dimension-adaptive tensor-product quadrature. *Computing*, 71(1):65–87, 2003. [31](#)
- [34] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. User’s guide for NPSOL (Version 4.0): A Fortran package for nonlinear programming. Technical Report TR SOL-86-2, System Optimization Laboratory, Stanford University, Stanford, CA, 1986. [13](#)
- [35] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, San Diego, CA, 1981. [36](#)
- [36] A. A. Giunta and M. S. Eldred. Implementation of a trust region model management strategy in the DAKOTA optimization toolkit. In *Proc. 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, number AIAA-2000-4935, Long Beach, CA, September 6–8, 2000. [45](#)
- [37] G. H. Golub and J. H. Welsch. Calculation of gauss quadrature rules. *Mathematics of Computation*, 23(106):221–230, 1969. [20](#)
- [38] A. Haldar and S. Mahadevan. *Probability, Reliability, and Statistical Methods in Engineering Design*. Wiley, New York, 2000. [7](#), [8](#), [13](#)
- [39] M. Hohenbichler and R. Rackwitz. Sensitivity and importance measures in structural reliability. *Civil Eng. Syst.*, 3:203–209, 1986. [44](#)
- [40] M. Hohenbichler and R. Rackwitz. Improvement of second-order reliability estimates by importance sampling. *J. Eng. Mech., ASCE*, 114(12):2195–2199, 1988. [11](#), [12](#)
- [41] H.P. Hong. Simple approximations for improving second-order reliability estimates. *J. Eng. Mech., ASCE*, 125(5):592–595, 1999. [11](#), [12](#)
- [42] S. Hosder, R. W. Walters, and M. Balch. Efficient sampling for non-intrusive polynomial chaos applications with multiple uncertain input variables. In *Proceedings of the 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, number AIAA-2007-1939, Honolulu, HI, April 23–26, 2007. [28](#)
- [43] D. Huang, T. T. Allen, W. I. Notz, and N. Zeng. Global optimization of stochastic black-box systems via sequential kriging meta-models. *Journal of Global Optimization*, 34:441–466, 2006. [14](#), [16](#)
- [44] D. Jones, M. Schonlau, and W. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998. [13](#), [14](#), [16](#)
- [45] A. Karamchandani and C. A. Cornell. Sensitivity estimation within first and second order reliability methods. *Struct. Saf.*, 11:95–107, 1992. [44](#)
- [46] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. Prentice–Hall, 1974. [39](#)
- [47] J. C. Meza, R. A. Oliva, P. D. Hough, and P. J. Williams. OPT++: an object oriented toolkit for nonlinear optimization. *ACM Transactions on Mathematical Software*, 33(2), 2007. [13](#)

- [48] F. Nobile, R. Tempone, and C. G. Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. Technical Report Technical report TRITA-NA 2007:7, Royal Institute of Technology, Stockholm, Sweden, 2007. [26](#)
- [49] F. Nobile, R. Tempone, and C. G. Webster. An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM J. on Num. Anal.*, 46(5):2411–2442, 2008. [26](#)
- [50] J. Nocedal and Wright S. J. *Numerical Optimization*. Springer Series in Operations Research. Springer, New York, 1999. [36](#), [49](#)
- [51] E. O. Omojokun. *Trust Region Algorithms for Optimization with Nonlinear Equality and Inequality Constraints*. PhD thesis, University of Colorado, Boulder, Colorado, 1989. [37](#)
- [52] V. M. Pérez, M. S. Eldred, , and J. E. Renaud. Solving the infeasible trust-region problem using approximations. In *Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, NY, Aug. 30–Sept. 1, 2004. AIAA Paper 2004-4312. [37](#), [39](#)
- [53] V. M. Pérez, J. E. Renaud, and L. T. Watson. An interior-point sequential approximation optimization methodology. *Structural and Multidisciplinary Optimization*, 27(5):360–370, July 2004. [36](#), [37](#), [39](#), [41](#)
- [54] R. Rackwitz. Optimization and risk acceptability based on the Life Quality Index. *Struct. Saf.*, 24:297–331, 2002. [44](#)
- [55] R. Rackwitz and B. Fiessler. Structural reliability under combined random load sequences. *Comput. Struct.*, 9:489–494, 1978. [8](#)
- [56] P. Ranjan, Bingham D., and Michailidis G. Sequential experiment design for contour estimation from complex computer codes. *Technometrics*, 50(4):527–541, 2008. [17](#)
- [57] M. T. Reagan, H. N. Najm, P. P. Pebay, O. M. Knio, and R. G. Ghanem. Quantifying uncertainty in chemical systems modeling. *Int. J. Chem. Kinet.*, 37:368–382, 2005. [29](#)
- [58] J. F. Rodriguez, J. E. Renaud, and L. T. Watson. Convergence of trust region augmented lagrangian methods using variable fidelity approximation data. *Structural Optimization*, 15:1–7, 1998. [35](#)
- [59] M. Rosenblatt. Remarks on a multivariate transformation. *Annals of Mathematical Statistics*, 23(3):470–472, 1952. [8](#), [23](#)
- [60] J. Sacks, S. B. Schiller, and W. Welch. Design for computer experiments. *Technometrics*, 31:41–47, 1989. [15](#)
- [61] I.C. Simpson. Numerical integration over a semi-infinite interval, using the lognormal distribution. *Numerische Mathematik*, 31:71–76, 1978. [20](#)
- [62] S.A. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Dokl. Akad. Nauk SSSR*, 4:240–243, 1963. [26](#)
- [63] A. Stroud. *Approximate Calculation of Multiple Integrals*. Prentice Hall, 1971. [27](#)
- [64] G. Tang, G. Iaccarino, and M. S Eldred. Global sensitivity analysis for stochastic collocation expansion. In *Proceedings of the 51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference (12th AIAA Non-Deterministic Approaches conference)*, Orlando, FL, April 12-15, 2010. AIAA Paper 2010-XXXX. [30](#)
- [65] M.A. Tatang. *Direct incorporation of uncertainty in chemical and environmental engineering systems*. PhD thesis, MIT, 1995. [28](#)

- [66] J. Tu, K. K. Choi, and Y. H. Park. A new study on reliability-based design optimization. *J. Mech. Design*, 121:557–564, 1999. [9](#)
- [67] G. N. Vanderplaats. *Numerical Optimization Techniques for Engineering Design: With Applications*. McGraw-Hill, New York, 1984. [36](#), [38](#)
- [68] R. W. Walters. Towards stochastic fluid mechanics via polynomial chaos. In *Proceedings of the 41st AIAA Aerospace Sciences Meeting and Exhibit*, number AIAA-2003-0413, Reno, NV, January 6–9, 2003. [28](#)
- [69] G. W. Wasilkowski and H. Woźniakowski. Explicit cost bounds of algorithms for multivariate tensor product problems. *Journal of Complexity*, 11:1–56, 1995. [26](#)
- [70] N. Wiener. The homogeneous chaos. *Amer. J. Math.*, 60:897–936, 1938. [19](#)
- [71] J. A. S. Witteveen and H. Bijl. Modeling arbitrary uncertainties using gram-schmidt polynomial chaos. In *Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit*, number AIAA-2006-0896, Reno, NV, January 9–12 2006. [20](#)
- [72] Y.-T. Wu, Y. Shin, R. Sues, and M. Cesare. Safety-factor based approach for probability-based design optimization. In *Proc. 42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, number AIAA-2001-1522, Seattle, WA, April 16–19 2001. [45](#), [49](#)
- [73] Y.-T. Wu and P.H. Wirsching. A new algorithm for structural reliability estimation. *J. Eng. Mech.*, ASCE, 113:1319–1336, 1987. [8](#)
- [74] B. A. Wujek and J. E. Renaud. New adaptive move-limit management strategy for approximate optimization, part 1. *AIAA Journal*, 36(10):1911–1921, 1998. [37](#)
- [75] B. A. Wujek and J. E. Renaud. New adaptive move-limit management strategy for approximate optimization, part 2. *AIAA Journal*, 36(10):1922–1934, 1998. [37](#)
- [76] D. Xiu. Numerical integration formulas of degree two. *Applied Numerical Mathematics*, 58:1515–1520, 2008. [27](#)
- [77] D. Xiu and J.S. Hesthaven. High-order collocation methods for differential equations with random inputs. *SIAM J. Sci. Comput.*, 27(3):1118–1139 (electronic), 2005. [26](#)
- [78] S. Xu and R. V. Grandhi. Effective two-point function approximation for design optimization. *AIAA J.*, 36(12):2269–2275, 1998. [10](#)
- [79] T. Zou, S. Mahadevan, and R. Rebba. Computational efficiency in reliability-based optimization. In *Proceedings of the 9th ASCE Specialty Conference on Probabilistic Mechanics and Structural Reliability*, Albuquerque, NM, July 26–28, 2004. [45](#)
- [80] T. Zou, Z. Mourelatos, S. Mahadevan, and P. Meernik. Reliability analysis of automotive body-door subsystem. *Rel. Eng. and Sys. Safety*, 78:315–324, 2002. [14](#)